



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

**Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa**

Estudio del proceso de automatización de una parte de la línea de fabricación de vehículos automotores

Por Sergio Gutiérrez Avenozza

PROYECTO FINAL DE GRADO

Grado: Ingeniería Industrial Electrónica y Automática

Director: José Luís Romeral Martínez

Fecha: Mayo 2019

SUMARIO

1	OBJETIVOS Y RESUMEN	6
2	INTRODUCCIÓN A LOS DISPOSITIVOS PLC.....	8
	2.1 HISTORIA.....	9
	2.2 DESARROLLO.....	10
	2.3 FUNCIONES.....	11
	2.4 ESTRUCTURA INTERNA	12
	2.5 CICLO DE FUNCIONAMIENTO.....	13
3	EL LENGUAJE STEP 7	14
	3.1 FUNCIONES OB.....	20
	3.2 FUNCIONES FB	21
	3.3 FUNCIONES FC	21
	3.4 FUNCIONES DE SEGURIDAD	22
	3.5 BLOQUES DE DATOS DB.....	22
	3.6 FUNCIONES VAT.....	23
4	EL ENTORNO TIA PORTAL.....	23
5	ESPECIFICACIONES.....	24
	5.1 DESCRIPCIÓN DE LA INSTALACIÓN	25
	5.2 CPU S7-416F-3 PN/DP.....	27
	5.3 PERIFERIA DESCENTRALIZADA ET200.....	27
	5.4 PUPITRES.....	28
	5.5 HMI, VISUALIZACIÓN Y GESTIÓN	29
	5.6 REDES PROFINET Y ETHERNET	30
6	PROGRAMA STEP 7	33
	6.1 DESCRIPCIÓN DEL ENTORNO.....	33
	6.2 CONFIGURACIÓN DEL HARDWARE	34
	6.3 SÍMBOLOS DEL PROGRAMA.....	37
	6.4 BLOQUES DE FUNCIÓN OB.....	38
	6.5 BLOQUES DE FUNCIÓN FB.....	39
	6.5.1 Descripción del problema	39
	6.5.2 Metodología empleada	40
	6.5.3 Secuencia de programación	43
	6.6 BLOQUES DE FUNCIÓN FC.....	60

	6.7 BLOQUES DE DATOS DB.....	63
	6.8 BLOQUES DE SEGURIDAD.....	64
7	PROGRAMA TIA PORTAL.....	64
	7.1 IMÁGENES DEL PROYECTO	65
	7.1.1 Imagen 01-Layout Principal	65
	7.1.2 Imagen 02-Pupitre 02	67
	7.2 FACEPLATES.....	68
	7.2.1. Creación de Faceplates.....	69
8	ANÁLISIS DEL IMPACTO AMBIENTAL	71
9	VALORACIÓN ECONÓMICA DEL PROYECTO.....	74
10	CONCLUSIONES.....	78
11	BIBLIOGRAFÍA Y REFERENCIAS	80

ÍNDICE DE FIGURAS

Figura 1. PLC de SIEMENS.....	9
Figura 2. Relé de OMRON.....	9
Figura 3. Esquema de la estructura interna de un PLC [3].....	12
Figura 4. Ciclo de trabajo de un PLC [5]	15
Figura 5. Ejemplo de programación en KOP	17
Figura 6. Ejemplo de programación en SCL	17
Figura 7. Ejemplo de programación en FUP	18
Figura 8. Ejemplo de programación en GRAFCET.	19
Figura 9. Ejemplo de programación en AWL.....	19
Figura 10. Layout de la instalación con mesas de rodillos, giratorias y elevadoras	26
Figura 11. PLC S7-400F	28
Figura 12. Panel de control de un pupitre	29
Figura 13. Esquema de red Profinet y Ethernet	32
Figura 14. Entorno del programa	33
Figura 15. Árbol del proyecto	34
Figura 16. Desplegable de objetos en Simatic 400 Step7	34
Figura 17. Lista de elementos del hardware con direcciones IP de la instalación.	36
Figura 18. Ventana de configuración del hardware.	37
Figura 19. Ruta de acceso a la tabla de símbolos.....	37
Figura 20. Tabla de símbolos.....	38
Figura 21. Representación de un bloque FB en el entorno SIMATIC	40
Figura 22. Representación de la etapa de autorización de carga entre mesas en sentido directo	41
Figura 23. Representación de la etapa de petición de descarga entre mesas en sentido directo.....	42
Figura 24. Representación de fin de tránsito.....	42
Figura 25. Función FB MR_2D_2S	43
Figura 26. Reset temporales, señal siempre a 0 y a 1	45
Figura 27. Entradas de tránsitos de forma directa.....	46
Figura 28. Defecto de presencias	47
Figura 29. Resumen de fallos	48
Figura 30. Inicio de secuencia	48
Figura 31. Tránsitos de carga y descarga.....	49

Figura 32. Reinicio de los tránsitos	49
Figura 33. Activación Step Cycle 00	50
Figura 34. Sequence step 00	51
Figura 35. Sequence Step 01	52
Figura 36. Sequence Step 02 y 03.....	53
Figura 37. Sequence Step 04	54
Figura 38. Sequence Step 05 y 06.....	55
Figura 39. Autorización carga directa.....	55
Figura 40. Autorización de carga inversa	56
Figura 41. Petición de descarga en sentido directo.....	57
Figura 42. Petición de descarga en sentido inverso.....	57
Figura 43. Fin de tránsito directo e inverso	58
Figura 44. Salidas marcha rodillos rápida y lenta.....	59
Figura 45. Salida de resumen de fallos.....	60
Figura 46. Bloques de función FC representados en SIMATIC.....	60
Figura 47. Fallos en el FC de una mesa	61
Figura 48. Llamada a una FB dentro de una FC de mesa.....	62
Figura 49. Lista de DB del programa en el entorno de SIMATIC.....	63
Figura 50. Fragmento del bloque de datos de una FB	63
Figura 51. Bloques de seguridad del programa.....	64
Figura 52. Imagen del Layout principal de la instalación	65
Figura 53. Fallo en mesa	66
Figura 54. Ventana de lotificador por colores.....	67
Figura 55. Imagen del Pupitre 2.....	68
Figura 56. Ejemplo creación faceplate	69

1 OBJETIVOS Y RESUMEN

La automatización está tomando un papel importantísimo en el desarrollo de la actividad para la industria de la producción en cadena o en serie, ya que permite generar mucho más producto finalizado en menos tiempo y con mayor precisión. En este caso, el sector del automóvil se está beneficiando de los avances tecnológicos que propone las nuevas generaciones de autómatas. De este modo, se gana tiempo en el proceso y en conjunto con las estaciones robotizadas se llega a obtener un producto de mejor calidad, aumentando así el volumen de producción. Este hecho va ligado a una mejora de software y hardware constante. Por este motivo, y enfocándolo a lo que se va a exponer en el proyecto, la problemática principal es la necesidad de generar funciones más eficientes y obtener un programa acorde con las capacidades de los nuevos PLC que se van incorporando para el transporte del producto.

Por otro lado, está la necesidad de estandarizar la programación. Se puede observar que, en esta industria, se amortizan al máximo los sistemas que resultan competentes por la exigencia de su mercado. Esto ocasiona problemas porque se vuelven obsoletos y al modernizar una instalación entra en conflicto con el resto de las instalaciones antiguas. Así pues, la diferencia de años se vuelve insalvable para conseguir tener programas parecidos y accesibles del mismo modo. Aquí toma parte, también, la gran variedad en la oferta de marcas de PLC y los lenguajes propios de programación.

El objetivo de este proyecto, como se puede apreciar, está enfocado en la automatización del transporte en la industria del automóvil y se tratará la programación de un conjunto de PLC de mejores prestaciones, generando funciones estándar de acuerdo con los requisitos de la línea de transporte. La programación de los PLC se realizará a través de STEP 7 de SIEMENS. Por otro lado, para la programación de las pantallas de los paneles de operador HMI, se empleará el entorno TIA Portal, también de SIEMENS. En STEP 7 se llevarán a cabo una serie de funciones llamadas FB, que se utilizarán para

crear una única función a la que podamos llamar en distintas ocasiones para agilizar la programación. Se puede decir que se han generado funciones estándar según las necesidades que ofrecen las mesas de transporte de las carrocerías. En cuanto a los paneles HMI, que son de control y ayuda a mantenimiento, se realizará una programación de Faceplates en TIA Portal, que es similar a una FB.

Lo que se pretende con las FB o nuevas funciones es facilitar el desarrollo de código para las siguientes líneas a modernizar, partiendo así desde cero, con la finalidad de establecer un modelo. La intención es buscar una solución para las intervenciones de modernización a corto plazo en las próximas instalaciones, dejando atrás los antiguos modelos de PLC de otras marcas con sus respectivos programas y lenguajes. Esto es muy ventajoso tanto para los técnicos de desarrollo como para los que tendrán que acceder a lo largo de su vida útil, porque evitará una gran cantidad de horas de programación y un fácil entendimiento del código.

El objetivo del proyecto no es simplemente programar código, sino desarrollar una tecnología de módulos fácilmente exportable a otras instalaciones con el mismo lenguaje, y con algunos cambios estructurales, a otros autómatas. Por lo tanto, se trata de abordar un problema tecnológico mediante una solución de ingeniería.

En lo que respecta al alcance del proyecto, se empieza a trabajar a partir del pliego de condiciones que se propone y las limitaciones mecánicas que ofrece la línea. Como la actualización es puramente eléctrica y del autómata, hay que estudiar todos los elementos de campo ya sean: sensores, actuadores, variadores o motores y sobre todo el ciclo de trabajo de la instalación porque no debe variar. Al igual que el entorno de programación, en este caso el STEP7, que ya viene estipulado. Con esto se pretende dar a entender que el verdadero desarrollo de la ingeniería electrónica está en la elección y en la toma de decisiones con respecto a la estructura del programa, la gestión de la

memoria de la CPU, la utilización de los recursos del entorno de SIEMENS, la organización de las entradas/salidas y la capacidad de dar soluciones ante nuevas situaciones.

También, como parte de los requisitos, se especifican los dispositivos o hardware empleados, que en este caso se trata de un PLC de la gama S7-400 y de la versión de STEP7 que es la 5.5. A nivel de programa solo se ha tenido en cuenta la condición de mantener los bloques de FB dentro de bloques FC con las correspondientes averías para agilizar la búsqueda de fallos de las mesas en el código. Además, la comunicación y los tipos de redes por los cuales se distribuirá la instalación, son decisiones tomadas en conjunto con la ingeniería eléctrica para su óptimo rendimiento. En este caso se trata de distribución descentralizada por ET200M que da flexibilidad a la hora de cablear todos los elementos de campo.

En definitiva, en este proyecto se pone a prueba la capacidad de reinventar y compaginar tecnología de hace 30 años con nuevas ideas y técnicas. La finalidad es mejorar las condiciones de una instalación de la industria automovilística mediante la automatización enfocada al diseño de programa.

2 INTRODUCCIÓN A LOS DISPOSITIVOS PLC

Los PLC, cuyas siglas significan Programmable Logic Controller y también llamados autómatas programables, son dispositivos programables que tienen el fin de automatizar procesos electromecánicos. Se les puede calificar como computadoras porque todos llevan un microprocesador. Sin embargo, el término computadora no hace referencia a una computadora doméstica o PC, sino a una computadora que está construida generalmente para ser utilizada en la industria y soportar ambientes hostiles de trabajo. En la figura 1 se pueden observar distintos modelos de PLC de Siemens.

Estos dispositivos son muy utilizados en maquinaria y en procesos de líneas de montaje donde se produce de manera continua y se repiten las acciones cíclicamente. Por ello, el PLC es capaz de analizar una gran cantidad de señales, denominadas entradas, con el objetivo de captar la mayor información posible de cómo se encuentra el medio o el proceso que debe gobernar. Es decir, determina en cada momento el estado real de la instalación, máquina, etc. Todo esto para, posteriormente, dar respuesta en modo de salidas y, mediante una gestión de código que el programador o la ingeniería electrónica ha desarrollado, activar los elementos mecánicos deseados.



Figura 1. PLC de SIEMENS

2.1 HISTORIA

A finales de los años 60, surge la necesidad en la industria de encontrar nuevas tecnologías para poder substituir los sistemas de control basados en circuitos con relés o interruptores. Un relé no es más que una bobina que por magnetismo cierra o abre los contactos, como un interruptor que abre o cierra el flujo de corriente. En la figura 2 se muestra un ejemplo de relé de OMRON. La dificultad o desventaja que implicaba trabajar con este tipo de circuitos de relés radica en que, en el caso de que la función a desempeñar variara, era necesario diseñar otro circuito diferente, modificando físicamente la conexión de esos interruptores o relés. Hoy en día, la programación mediante dispositivos PLC



*Figura 2. Relé de
OMRON*

ofrece la posibilidad de variar las funciones, sin tener que modificar elementos físicamente, si cambiamos el proceso que estamos controlando.

En 1968 GM Hydramatic, la división de General Motors, pidió propuestas para ver si alguna empresa le daba soluciones al problema. Fue entonces cuando Bedford Associates diseñó el primer PLC llamado 084. A partir de ahí se creó la empresa Modicon (MOdularDIgitalCONtroller) para la comercialización de este nuevo producto. Dick Morley fue una de las personas que trabajó en ese proyecto por lo que es considerado el padre del PLC.

La marca Modicon ha sido traspasada desde entonces, a varias empresas conocidas como son Gould Electronics, la compañía alemana AEG y, por último, el actual propietario francés Schneider Electric [1].

2.2 DESARROLLO

Los primeros PLC se diseñaron con el objetivo de reemplazar los sistemas de relés lógicos. Estos PLC fueron programados en un lenguaje llamado “Listado de instrucciones”, con el que las órdenes de control se le indicaban al procesador en forma de un listado secuencial de códigos en lenguaje de máquinas. Más adelante, para tratar de facilitar el mantenimiento de los sistemas a controlar, se comenzó a utilizar un lenguaje gráfico llamado Ladder, más conocido como diagrama de escalera, el cual es muy similar a un diagrama esquemático de la lógica de relés. Se escogió este sistema para minimizar las demandas de formación de los técnicos existentes. Otros autómatas primarios utilizaron un formulario de listas de instrucciones de programación.

Actualmente, los PLC modernos ofrecen al usuario la posibilidad ser programados de diversas formas, desde diagramas de contactos hasta lenguajes de programación adaptados de BASIC y C [2].

2.3 FUNCIONES

El control del relé secuencial, el control de procesos, el control de movimiento, los sistemas de control distribuido y la comunicación por red son las funciones principales de los PLC. Estas funciones han ido evolucionando a medida que se ha incrementado la complejidad de los procesos productivos de la industria. Las capacidades de los dispositivos PLC en cuanto a almacenamiento, manipulación, potencia de procesamiento y de comunicación son prácticamente equivalentes a los ordenadores personales de sobremesa. Sin embargo, el hardware de los ordenadores de sobremesa no suele estar diseñado para soportar los mismos niveles de tolerancia a la temperatura, humedad o vibraciones, ni garantiza la longevidad de los procesadores utilizados en los PLC. Además, caben destacar sus limitaciones de una lógica basada en una interfaz de escritorio, con sistemas operativos como Windows, los cuales no se prestan a la ejecución de una lógica determinista. Un PLC siempre sigue la misma secuencia de ejecución y solo es posible modificarla mediante un cambio de código a través de un PC con accesibilidad al entorno de programación.

En cuanto a la viabilidad de los PCs, generalmente no se han aceptado en la industria pesada, ya que estos ordenadores ejecutan sistemas operativos menos estables que los PLC. No obstante, estas aplicaciones de escritorio encuentran uso en situaciones menos críticas como podrían ser la automatización de laboratorio y su uso en instalaciones pequeñas, en las cuales la aplicación es de menor exigencia y crítica ya que, por lo general, son dispositivos mucho menos costosos que los PLC.

2.4 ESTRUCTURA INTERNA

Las partes fundamentales que forman un PLC son la unidad central de proceso o CPU y las interfaces de entrada y salida. En la figura 3 se representa un esquema de la estructura interna de un PLC.

La CPU interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema del PLC, formada por el procesador y la memoria. El procesador tiene la función de ejecutar el programa generado por el usuario, el cual se encuentra almacenado en la memoria. Además, el procesador, realiza funciones de autodiagnóstico y se comunica con el exterior a través de sus puertos de comunicación.

La interfaz de entrada se encarga de adaptar las señales que provienen de los elementos captadores, como podrían ser botoneras, sensores o llaves a un nivel que el CPU pueda interpretar como información. La finalidad de la CPU es ejecutar el programa para activar las salidas. La interfaz de salida es la responsable de administrar la potencia necesaria para controlar el actuador.

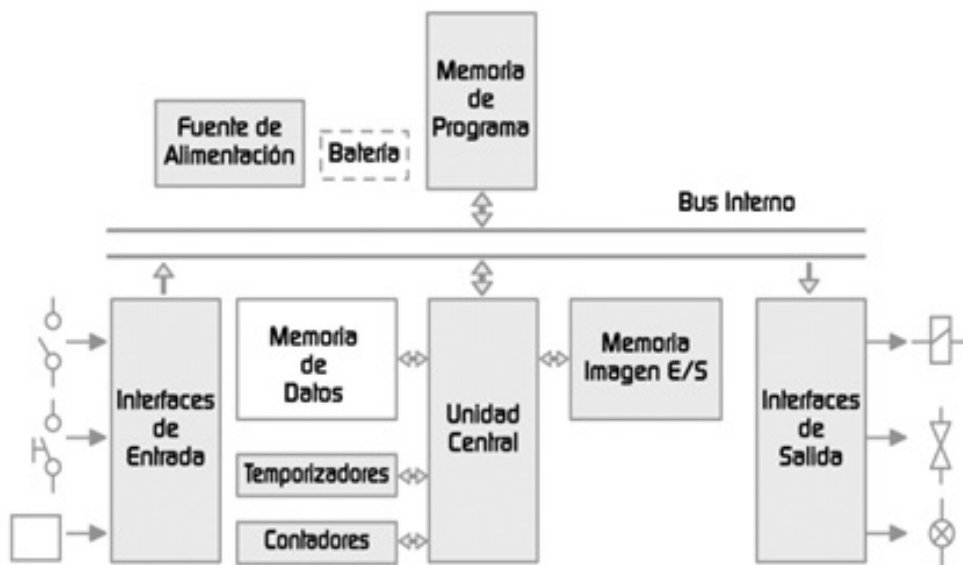


Figura 3. Esquema de la estructura interna de un PLC [3]

2.5 CICLO DE FUNCIONAMIENTO

El ciclo de funcionamiento de un autómatas se basa en un funcionamiento secuencial y cíclico, es decir, durante el tiempo en el que se encuentre alimentado, dispuesto en modo RUN y sin errores, ejecutará una serie de operaciones, entre las cuales se encuentra el programa realizado por el usuario, de forma continua. En la figura 4 aparece un diagrama de flujo que representa el ciclo de funcionamiento de un autómatas. Como se muestra en la figura, el funcionamiento de un PLC se divide en dos partes: ciclo inicial, y ciclo de operación.

Durante el ciclo inicial, el autómatas realiza una serie de acciones que tratan fundamentalmente de inicializar los estados del mismo, y chequear el hardware (conexiones, batería, memoria, etc.).

Una vez superado el ciclo inicial, el autómatas entra en el ciclo de operación, que se divide en tres bloques:

- Proceso común: se realiza el chequeo cíclico de conexiones (tensión, batería y buses de conexión con las interfaces), así como el chequeo de la memoria de programa, para comprobar la integridad de la misma y los posibles errores de sintaxis.
- Ejecución del programa: se consultan los estados de las entradas y las salidas y se elaboran órdenes de mando a partir de ellos.
- Servicio a periféricos: este último bloque únicamente se realiza cuando es necesario algún intercambio de datos entre el PLC y algún equipo periférico [4].

3 EL LENGUAJE STEP 7

El lenguaje STEP 7 es un software de programación para los PLC de la familia SIMATIC-S7 de Siemens. Este lenguaje es el sucesor de S5, el cual está ampliamente extendido en Alemania. Los autómatas de SIMATIC constituyen un estándar que compite en primera línea con otros sistemas de programación y control lógico de autómatas [5]. El desarrollo del estándar de programación, que se explicará en este proyecto, está hecho en STEP 7.

STEP 7 es una de las herramientas de ingeniería más modernas para llevar a cabo la configuración y programación de todos los controladores de la familia SIMATIC. Para las tareas más sencillas de visualización, con los SIMATIC Basic Panels, se incluye SIMATIC WinCC Basic. Entre las principales ventajas que nos ofrece este lenguaje podemos destacar:

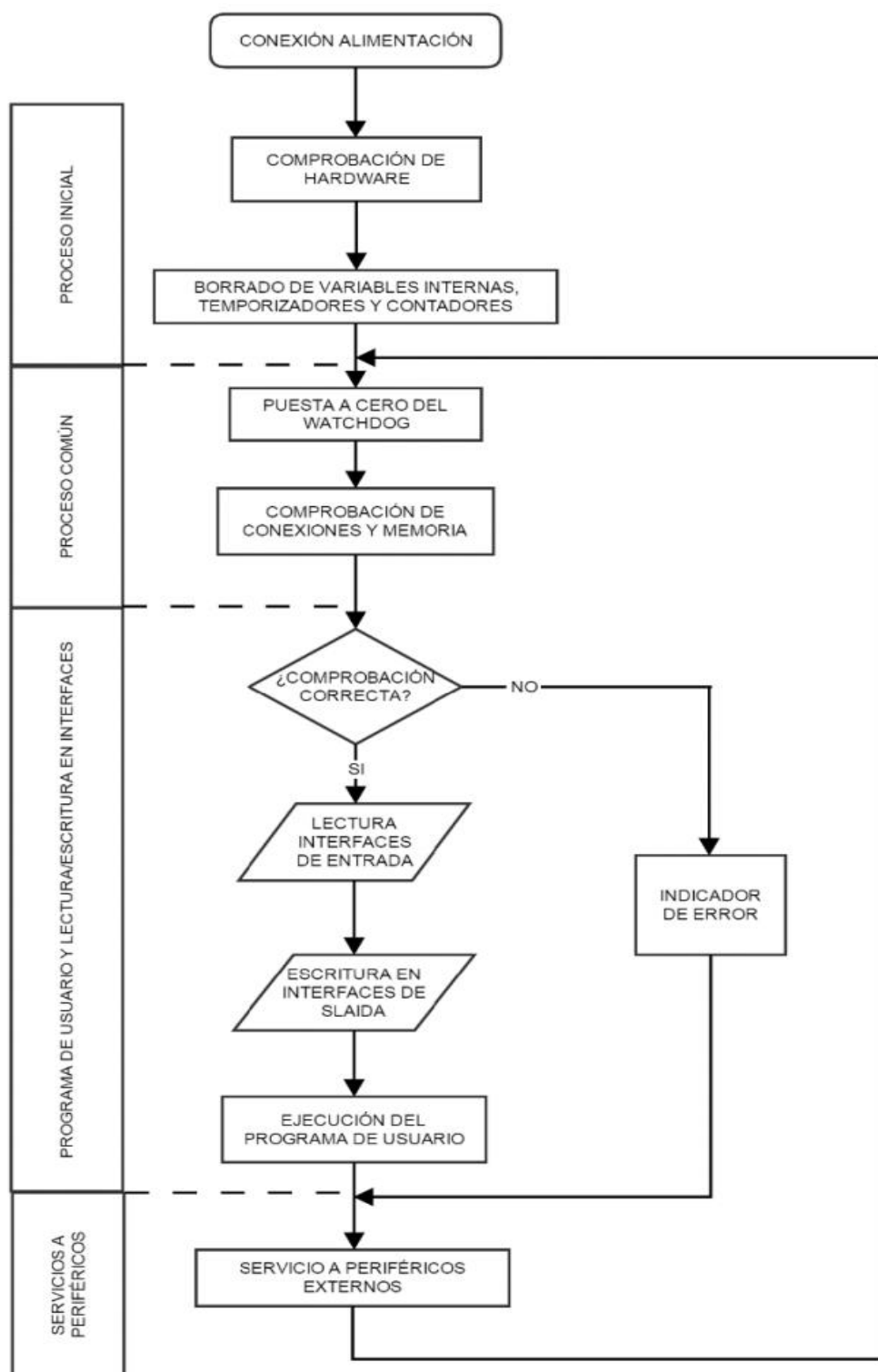


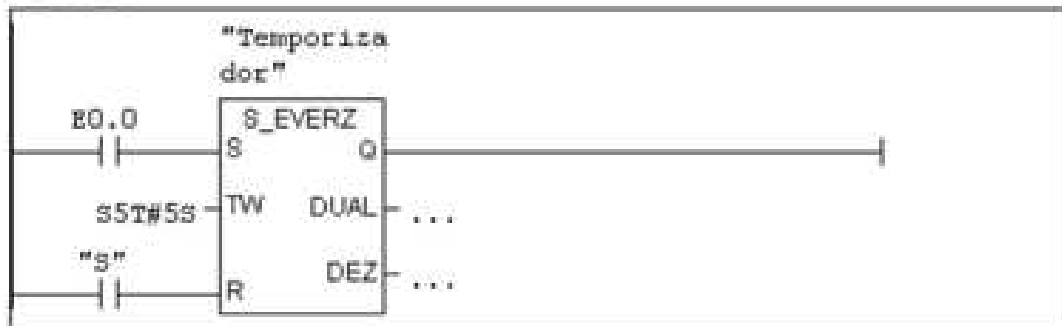
Figura 4. Ciclo de trabajo de un PLC [5]

- Se reduce el tiempo de desarrollo de la ingeniería gracias a las innovaciones de las nuevas funciones integradas y su configuración gráfica.
- Rapidez en las puestas en marcha, localización de errores eficiente mediante el diagnóstico del sistema integrado. Esto sucede por el alto rendimiento de seguimiento en tiempo real junto con el amplio abanico de funciones en línea.
- Menor tiempo de parada, gracias a un mantenimiento a distancia sencillo y al diagnóstico con el servidor web y Teleservice.
- Seguridad de las inversiones por la reutilización de componentes, el uso de librerías y la compatibilidad de versiones.

Mediante las funciones “drag and drop”, que trata de copiar y pegar segmentos de código fácilmente, se agiliza y se facilita la programación con este software. Los principales lenguajes de programación dentro de este entorno, de alto y bajo nivel, son:

Esquema de contactos (KOP): conocido como diagrama de contactos o de escalera. Probablemente el más extendido entre todos los lenguajes de programación. De forma similar a los esquemas de circuitos, encontramos elementos como los contactos normalmente cerrados y normalmente abiertos que, junto con una gran diversidad de funciones, se agrupan en segmentos. Uno o varios segmentos forman el área de instrucciones de un bloque lógico. Es lenguaje de nivel alto [6]. En la figura 5 se representa un ejemplo de temporizador programado en lenguaje KOP.

Segm. 1: Marcha temporizador



Segm. 2: Marcha motor



Figura 5. Ejemplo de programación en KOP

Texto estructurado (SCL): es un lenguaje estructurado de control basado en texto diseñado para escribir código de forma rápida y eficaz. Entre sus principales ventajas se puede destacar que permite utilizarse para generar documentos equivalentes en TeX, docbook, HTML u otros lenguajes. Es lenguaje de nivel alto [7]. En la figura 6 se observa un fragmento de programación en SCL.

```
//Activar una salida (Set)
IF "Entrada1"=true THEN
    "Salida1" := true;
END_IF;
//Desactivar una salida (Reset)
IF "Entrada2"=true THEN
    "Salida1" := false;
END_IF;
```

Figura 6. Ejemplo de programación en SCL

Diagrama de funciones (FUP): es un lenguaje gráfico que utiliza cuadros de álgebra booleana para la representación de su lógica. Permite al usuario representar funciones complejas mediante cuadros lógicos, ver de forma agrupada por bloques las diferentes lógicas y tener bloques complejos. Para situaciones donde nos encontremos con mucha programación en serie, suele ser la opción más compacta y más sencilla para ver el segmento al completo. Es un ejemplo de programación en alto nivel y se puede ver una muestra en la figura 7.

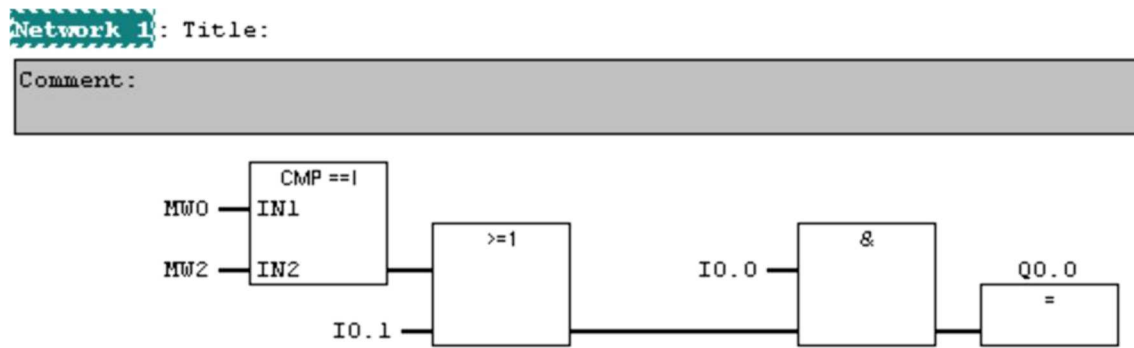


Figura 7. Ejemplo de programación en FUP

Programación secuencia (GRAPH): conocido como GRAFCET, es un lenguaje de representación gráfica de los comportamientos consecutivos de un sistema lógico que ha sido predefinido por sus entradas y salidas. Al mismo tiempo, es un diagrama funcional normalizado que permite al usuario realizar un modelo del proceso a automatizar, en el cual se contemplan las entradas, las acciones a realizar por el programa y los procesos intermedios que provocan la ejecución de estas acciones. Este es otro ejemplo de programación en alto nivel y se puede ver una representación en la figura 8.

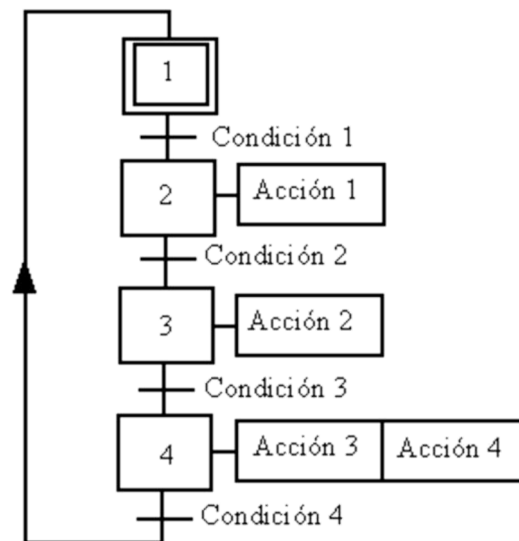


Figura 8. Ejemplo de programación en GRAFCET.

Lista de instrucciones (AWL): es un lenguaje textual orientado a la máquina. Con este lenguaje, las instrucciones equivalen en gran parte, a los pasos que la CPU sigue para la ejecución del programa. Es el lenguaje más completo y, al mismo tiempo, el más complejo visualmente de seguir y susceptible a errores [8]. Este es el ejemplo más claro de programación en nivel bajo. En la figura 9 se muestra un ejemplo de un fragmento en AWL.

```
OB1 : Control de motor
Segm. 1: Marcha temporizador
    U    E    0.0
    L    S5T#5S
    SE   T    0
    U    A    2.0
    R    T    0
    NOP  0
    NOP  0
    NOP  0
```

Figura 9. Ejemplo de programación en AWL

El Software STEP 7, además, ofrece paquetes opcionales S7-PLCim, S7-SCL, S7-GGRAPH, documentación de instalaciones y Teleservice de STEP 7 V5.5, todos estos integrados en STEP 7 Professional (TIA Portal). No sería necesaria ninguna licencia adicional.

El propietario del servicio de actualización del software STEP 7 Professional recibe automáticamente SIMATIC STEP 7 Professional (TIA Portal). [9]

3.1 FUNCIONES OB

Las funciones OB son unos bloques que ayudan al usuario a tener una programación mejor estructurada, optimizar los segmentos de código y, además, acceder a otras herramientas de los dispositivos PLC. Podemos encontrar diversos tipos de bloques, los cuales reaccionan a un evento específico en el PLC y tienen la capacidad de interrumpir el programa del usuario. Existen bloques de alarmas horarias, de interrupción de fallo, de interrupción de hardware y de fallo de ejecución del programa, entre otros. Las funciones OB se podrían definir de forma breve como unos bloques que ofrecen la oportunidad de interrumpir el PLC en caso de que se produzca algún tipo de fallo o cuando suceda un evento y sea necesario pausar el programa. En los PLC de Siemens podemos hallar una gran diversidad de bloques organizacionales OB, entre los que podríamos mencionar [10]:

- Bloque estándar para la elaboración cíclica del programa de usuario (OB1).
- OB de alarma horaria (OB10 a OB17).
- OB de alarma retardado (OB20 a OB23).
- OB de tiempo (OB30 a OB38).
- OB de alarma del proceso (OB40 a OB47).

- OB de alarma de estado (OB55).
- OB de alarma de actualización (OB56).

3.2 FUNCIONES FB

Los bloques de función FB son bloques programables y a diferencia de los OBs, disponen de un bloque de datos asignado como memoria. Este recibe el nombre de bloque de datos de instancia. Los parámetros que se transfieren a las FB y las variables estáticas se memorizan en el DB de instancia. Las variables temporales se memorizan en la pila de datos locales.

Los datos que se memorizan en el DB de instancia no se pierden una vez concluido el tratamiento de las FB. Los datos que se memorizan en la pila de datos locales sí que se pierden al concluir el tratamiento de las FB.

Estas FB simplifican la programación de funciones complejas de uso frecuente. Para ejecutar un programa que está desarrollado dentro de una FB, se necesita ejecutar la propia FB llamándola desde otro bloque de programación.

3.3 FUNCIONES FC

Las funciones FC también son bloques programables, pero estos no disponen de memoria. Los datos se van almacenando en la pila de datos locales y, una vez se ha terminado de usar la función, se pierden. En cualquier caso, si se pretende memorizar datos, las funciones FC tienen la posibilidad de utilizar bloques de datos globales.

En una FC se han de indicar siempre los parámetros actuales porque no tiene asignada ninguna memoria. A los datos locales de una FC no se le pueden asignar valores iniciales.

Estas funciones contienen un código o programa que se ejecutará cada vez que la FC sea llamada por otro bloque lógico. Se pueden utilizar para devolver un valor de función al bloque que lo invoca y para ejecutar el código desarrollado. [11]

3.4 FUNCIONES DE SEGURIDAD

Las funciones de seguridad se programan con las mismas herramientas que el resto de las funciones, anteriormente explicadas.

La generación del programa de seguridad, habitualmente se lleva a cabo en los lenguajes de programación FUP o KOP. Además, se dispone de una librería integrada con diversos bloques de función certificados por TÜV (empresa alemana de certificación de seguridad en fábricas y empresas), los cuales permiten materializar fácilmente funciones de seguridad. El sistema de librerías soporta la estandarización a nivel interno de la empresa y también ofrece una simplificación de la validación de las aplicaciones de seguridad [12].

3.5 BLOQUES DE DATOS DB

Los bloques DB (Data Blocks) en Step 7 son unas herramientas muy potentes que se utilizan para el almacenamiento y tratamiento de datos. Los DBs pueden ser:

- Globales: el acceso a los DB globales está diseñado para que se pueda acceder desde cualquier parte del programa, indistintamente de si el bloque que llama a esta función es del tipo OB, FC o FB.
- De instancia: los DB de instancia se asocian a los Bloques de función FB y almacenan los valores de las variables estáticas usadas en estas FB [13].

3.6 FUNCIONES VAT

Las funciones VAT son tablas de variables, donde el usuario puede crear las distintas variables que va a utilizar a lo largo del programa. También permiten ver el estado en el que se encuentran con el modo “observar” e incluso forzar sus valores para realizar una simulación [14].

4 EL ENTORNO TIA PORTAL

TIA Portal (Totally Integrated Automation Portal) es el software que ha sido utilizado en este proyecto de automatización para la generación de las HMI. Se trata de una plataforma de ingeniería que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción, ya que ofrece un entorno unificado para la realización de todas las tareas de control, visualización y accionamiento. Se caracteriza por tener una interfaz de usuario intuitiva, funciones sencillas y transparencia de datos, con lo cual es una herramienta muy fácil de utilizar.

TIA Portal utiliza un sistema de pantalla partida que ofrece la posibilidad de tener abiertos varios editores al mismo tiempo, lo cual permite el intercambio de datos entre los mismos.

Otra de las cualidades de TIA Portal es que solo es necesario introducir los datos que se van a emplear en los distintos editores una única vez. Gracias a la gestión de datos centralizada, los datos de aplicación que sean modificados a posteriori se actualizarán de forma automática para todos los equipos involucrados, tanto en los PLC como en los HMI. Esta gestión centralizada de los datos presenta varias ventajas:

- Ofrece soluciones reutilizables, ya que todos los datos pueden ser reutilizados en cualquier momento, tanto dentro de un proyecto como en librerías que pueden ser empleadas en otros proyectos.

- Se reduce considerablemente la probabilidad de que surjan errores, creando proyectos transparentes y compactos.

Además, los datos y proyectos preexistentes pueden integrarse sin ningún esfuerzo, lo cual asegura su inversión a largo plazo [15].

El TIA Portal incorpora las últimas versiones de Software de Ingeniería SIMATIC STEP 7, WinCC y Startdrive para la planificación, programación y diagnóstico de todos los controladores SIMATIC, pantallas de visualización y accionamientos SINAMICS de última generación.

El motivo principal por el cual se ha realizado el diseño gráfico de las HMI empleando TIA Portal en lugar de la herramienta equivalente en STEP 7, llamada WinCC, es debido a las características anteriormente comentadas: es más flexible, más intuitivo y ofrece una mayor cantidad de recursos de forma más sencilla, lo cual agiliza mucho el trabajo.

5 ESPECIFICACIONES

Llegados a este punto y después de haber sentado las bases de las herramientas para la programación, se entrará de lleno en el estudio del proyecto. Se empezará con una breve descripción de la instalación donde se aplicará el nuevo estándar de programación. Se continuará con la explicación de los bloques de programación diseñados para agilizar y mejorar la lógica empleada en STEP 7. Por último, se analizarán las pantallas HMI diseñadas para el usuario en TIA Portal V13.

5.1 DESCRIPCIÓN DE LA INSTALACIÓN

El principal cometido de la instalación, a la que se le han introducido los cambios de software y hardware, es el almacenaje de carrocerías. Las carrocerías se transportan encima de unos SKIDs (patín donde van montados los coches) que a su vez se mueven por las mesas de transporte. Estas están formadas por rodillos que giran con uno o varios motores según su movilidad o funcionalidad. Hay diferentes tipos como por ejemplo mesas de un sentido, mesas de dos sentidos, mesas giratorias o mesas elevadoras. En la figura 10 se muestra el layout de la planta y la disposición de todas las mesas de cada tipología.

Gracias al movimiento de los rodillos, los SKIDs junto con las carrocerías, se desplazan por las mesas hasta las posiciones deseadas. La prioridad principal de las nuevas funciones realizadas para el nuevo estándar de programación tiene que ver con la funcionalidad de cada mesa y la repetición en la instalación de cada tipo. Cada zona de la instalación está controlada por pupitres donde se pueden controlar manualmente las mesas. También se dispone de una pantalla de 15 pulgadas en el armario eléctrico principal y otra pantalla de 9 pulgadas en uno de los pupitres. Todo esto para poder visualizar el proceso de transporte.

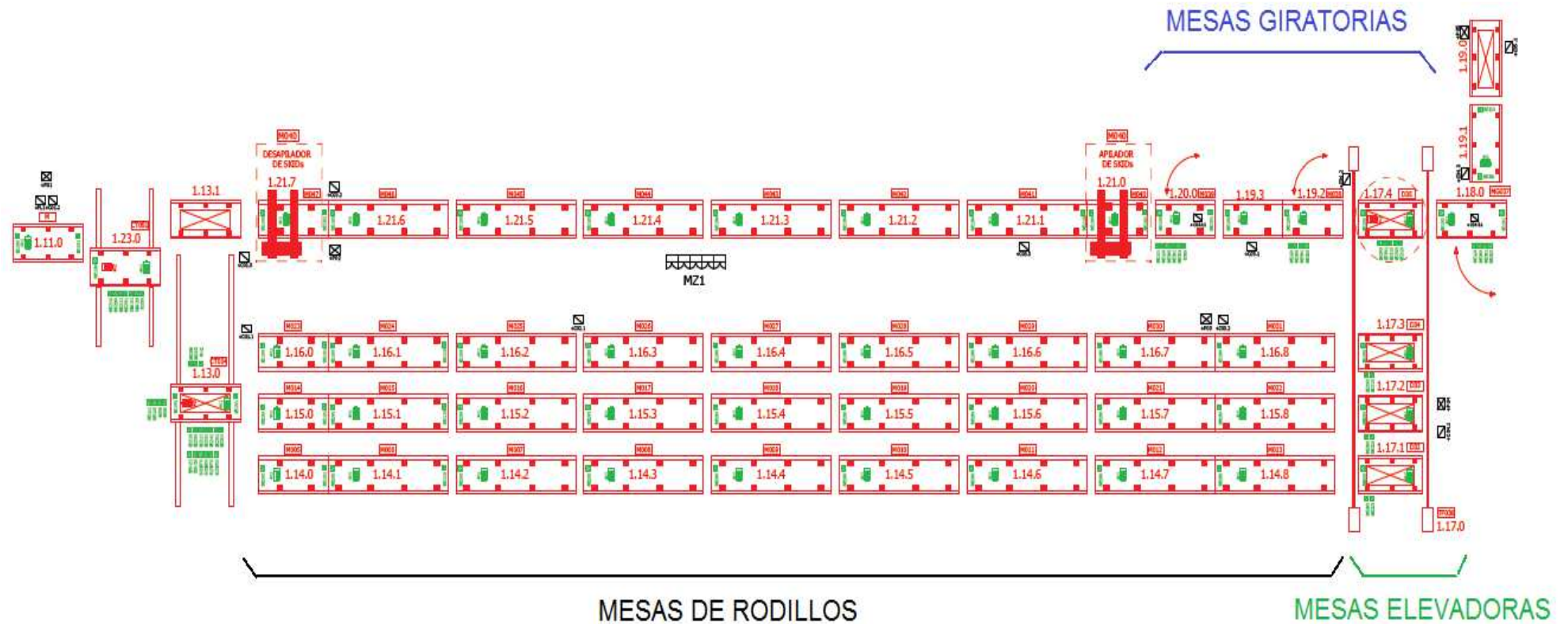


Figura 10. Layout de la instal·lació amb meses de rodillos, giratorias y elevadoras

El control del armario eléctrico de la instalación se gestionará con un S7-416 3F PN/DP, con una tarjeta de memoria RAM adicional de 4MB y con conexión a la red Ethernet Industrial interna mediante Switch Scalance XM416-4C y CP 443-1 y periferia descentralizada de I/O en S7 mediante red Profinet + Profisafe.

Se utilizará para el bus de campo Profinet, periferia con módulos ET200M en armario donde el módulo de interface es un IM 153-4 PN HF y con módulos ET200S en campo donde el módulo de interface es un IM151-3 PN HF, así como Scalance X-208 como maestros del anillo ProfiNet entre los módulos ET200M y ET200S. Se instalarán nuevos pupitres de control y cajas descentralizadas de campo controladas por periferia descentralizada I/O en S7 mediante Profinet+ Profisafe.

5.2 CPU S7-416F-3 PN/DP

La CPU S7-416F, que pertenece a la familia de las 400F de SIMATIC, está fabricada para instalaciones con unos requisitos del rango de potencia superiores. Gracias a su estructura modular, permite la ampliación de forma descentralizada con la periferia ET200M y ET200S. Disponen de una memoria RAM de 11.2 MB, MPI/PROFIBUS DP-Master-Interface, slot para tarjetas de memoria, PROFIBUS DP-Interface y PROFINET Interface. En la figura 11 se muestra una fotografía del PLC empleado.

5.3 PERIFERIA DESCENTRALIZADA ET200

Todos los elementos de la instalación, tanto los elementos de campo como los elementos de armario, se encuentran conectados mediante un sistema de periferia descentralizada o ET200. La periferia descentralizada se utiliza para no tener que cablear directamente las entradas y las salidas del PLC desde cualquier punto de la instalación. Cada elemento se cablea a la ET200 y cada una de esas ET200 se conecta al PLC mediante bus tipo Profinet. Las ET200S las podemos encontrar en cualquier punto de la instalación, para los elementos

de campo, y las ET200M se encuentran en el armario eléctrico. Ambas carecen de procesador, su función es hacer de cabecera.



Figura 11. PLC S7-400F

5.4 PUPITRES

Los pupitres son puntos de control y están repartidos por toda la instalación. Un pupitre se encarga de controlar un conjunto de mesas limitado y se compone de un panel de control y/o una HMI. En este panel de control, se puede seleccionar la mesa y accionarla manualmente según su funcionalidad. Esto quiere decir, por ejemplo, que si la mesa es elevadora se podrá mover arriba y

abajo. Si la mesa es giratoria se podrá rotar hacia la derecha o la izquierda. En todos los casos los rodillos se podrán mover hacia adelante y hacia atrás si el motor lo permite. Podemos ver el ejemplo en la figura 12. A parte de poder controlar las mesas, los pupitres disponen de selectores para cambiar de modo automático a manual, seta de emergencia, paro/marcha y prueba lámparas. La manera de numerar los pupitres es P01 para el pupitre número uno, P02 para el dos y así sucesivamente hasta 5 de los que se dispone.



Figura 12. Panel de control de un pupitre

5.5 HMI, VISUALIZACIÓN Y GESTIÓN

Las pantallas instaladas y programadas para la visualización y gestión de los procesos son:

- HMI TP1500 Confort, con Micro Memory Card para guardar backup MMC 2GB en el armario principal.
- HMI TP 900 Confort con Micro Memory Card para guardar backup MMC 2GB en el pupitre P01, para la monitorización y gestión de la validación, puesto de lectura y edición de SKIDs.

Estas pantallas se podrán editar con el software WinCC confort V13 Sp1 o superior.

Desde la TP1500 se ha de poder visualizar y controlar toda la instalación. Es decir, se podrán visualizar tanto alarmas como fallos que se produzcan en las mesas a lo largo del proceso del transporte de las carrocerías por la instalación. Como por ejemplo averías de los motores o si la mesa está ocupada por alguna carrocería.

En la TP900 confort del pupitre P01, se debe poder validar manualmente los fallos de lectura de los SKID que a la entrada del transporte no hayan podido ser leídos correctamente de forma automática. También se ha de disponer de una pantalla de edición de SKIDs para introducir datos de la carrocería que se desee. También en la TP900 se debe visualizar los grupos funcionales del proceso de las instalaciones que controlan, así como las alarmas, de igual manera que en la HMI TP1500.

5.6 REDES PROFINET Y ETHERNET

Para la red Profinet, se realizan ruteados de mangueras de cable Profinet, canalizaciones y elementos para la integración S7 en red (IE FC Outlet RJ45, Scalance X-208, Latiguillos RJ45-RJ45, certificación de red Profinet, etc.)

La instalación se ha realizado siguiendo el manual del buen instalador de la firma SIEMENS “PROFINET Installation Guideline for Commissioning” en el que se establecen las normas de montaje y configuración para este tipo de redes, con el fin de garantizar su correcto funcionamiento.

Se ha instalado un scalance como maestro del anillo, de 8 tomas RJ45 de Siemens Scalance X208 (Ref: 6GK5208-0BA10-2AA3), para garantizar la funcionalidad de la red Profinet en anillo. La topología del bus de campo Profinet para los módulos ET200S se ha realizado con estructura en anillo para evitar que un fallo de bus afecte a toda la instalación. Se ha configurado de manera que, en caso de que se pierda una conexión, continúe la comunicación

con la otra rama del anillo.

La topología de esta red se ha sido canalizada separándola de la red eléctrica, con canal o tubo galvanizado. La distancia máxima entre elementos Profinet no supera en ningún caso los 90m, por lo que si el recorrido que debe seguir el enlace Profinet superara esta distancia, se ha intercalado, en caja acristalada, un Switch Scalance entre el origen y el destino.

En cuanto a la red Ethernet, se han ruteado mangueras de cable Ethernet para bajante (en fibra o cobre), canalizaciones y elementos para la integración S7. Para la conexión del PLC a la red interna de la empresa formada por Scalance XM416-4C se ha instalado en el armario principal otro Scalance XM416-4C para su conexión mediante cobre o fibra.

La realización del bajante desde el armario de la instalación hasta el Scalance de la red Ethernet Industrial interna, se ha realizado por encima de las cerchas de la empresa, para evitar interferencias con instalaciones o desmontaje de otras instalaciones.

En la figura 13 aparece el esquema de red de las redes Profinet y Ethernet correspondiente a la instalación.

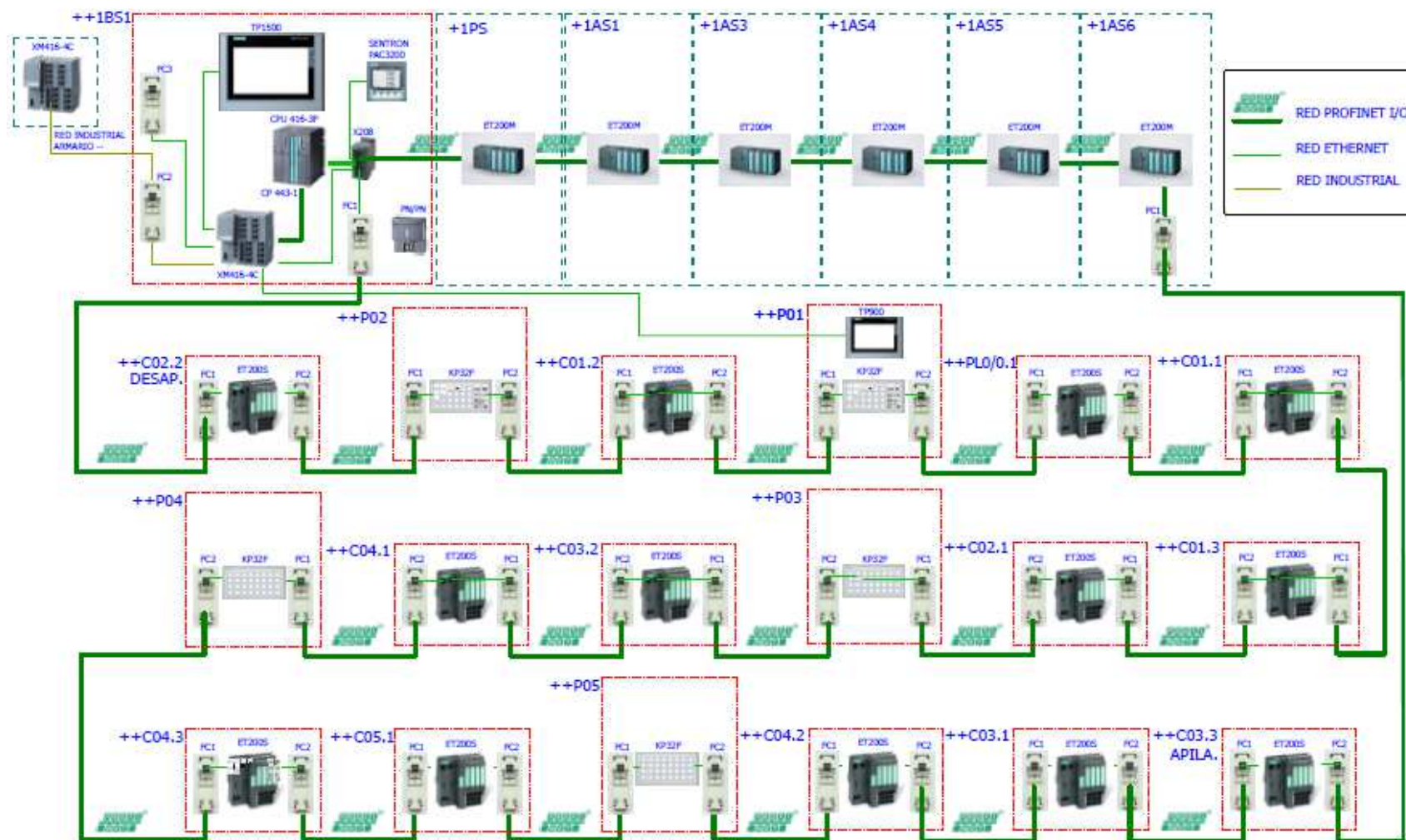


Figura 13. Esquema de red Profinet y Ethernet

6 PROGRAMA STEP 7

6.1 DESCRIPCIÓN DEL ENTORNO

Para llevar a cabo la realización de un proyecto con autómatas programables, el software de Siemens permite gestionar los datos empleados en la ejecución de cualquier programa. Mientras que la parte de la parametrización de los datos la tiene que realizar el usuario, la parte de aplicación directa de funciones la lleva integrada el software de forma automática.

El entorno o lo que se visualiza con la herramienta de Simatic Manager del software STEP 7 se muestra en la figura 14.

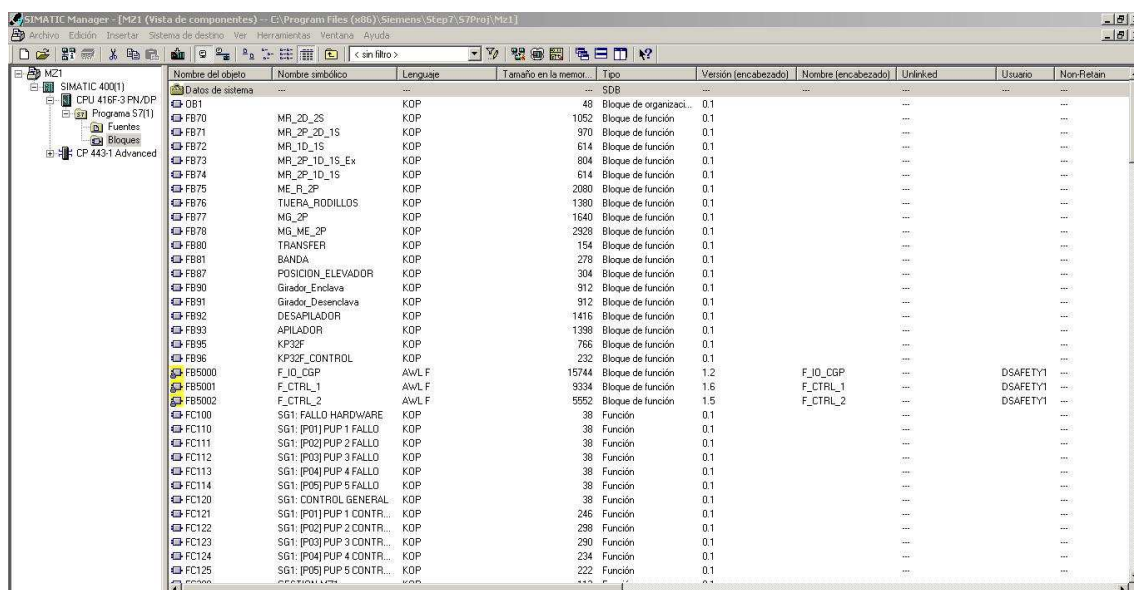


Figura 14. Entorno del programa

En primer lugar, se analizarán las distintas partes que forman este software en lo que se refiere a la programación y configuración de cada una de estas. En el interior del programa, en la parte izquierda, aparece el árbol del proyecto.

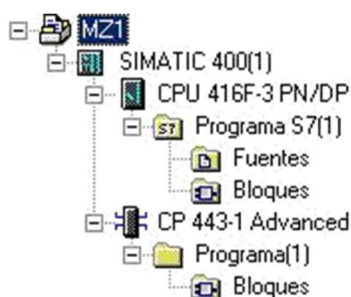


Figura 15. Árbol del proyecto

En la figura 16 se puede ver con mayor detalle el árbol del proyecto, en el cual se localizan la configuración del hardware del autómatas, los módulos de entradas y salidas que se ha adherido al sistema, los símbolos que se han utilizado para la programación y los bloques de funciones y datos que contienen dicha programación.

Según como se ha comentado en las especificaciones, se ha incluido en el proyecto el autómatas de la gama S7-400F con la CPU S7-416F-3 PN/DP y una CP 443-1 Advanced.

6.2 CONFIGURACIÓN DEL HARDWARE

Nombre del objeto	Nombre simbólico	Tipo
Hardware	---	Configuración del equipo
CPU 416F-3 PN/DP	---	CPU
CP 443-1 Advanced	---	CP

Figura 16. Desplegable de objetos en Simatic 400 Step7

En la figura 16, si se abre la pestaña de SIMATIC, se pueden ver los dos elementos principales físicos que conforman el autómatas. Estos son la CPU y la CP de comunicaciones. Para configurarlos se debe acceder a la opción hardware.

El primer paso para poder configurar el hardware es la introducción del bastidor o también llamado rack. En este punto, es posible parametrizar las principales propiedades de la CPU, entre las que podemos destacar el comportamiento del sistema tras un arranque en caliente, su tiempo de ciclo, etc.

Antes de configurar el hardware en el proyecto, es importante tener claro la cantidad de dispositivos que hay, su tipo, la parte de la instalación en la que se encuentran y la IP que se va a asignar a cada uno de estos elementos. La lista se muestra en la figura 17.

La dirección IP es un número que identifica, de manera lógica y jerárquica, a una interfaz en red de un dispositivo que utilice el protocolo IP o, que corresponde al nivel de red del modelo TCP/IP.

Una vez definidos todos los detalles de cada elemento del hardware, es posible pasar a introducir cada uno de estos en la pantalla de configuración del hardware de la forma que se visualiza en la figura 18. Mediante la representación gráfica de cada elemento se muestra la conexión a través de la red Profinet. La red está conectada al Puerto X5 de la CPU. Se ha distribuido por columnas, aunque el orden en el que aparecen estas columnas no tiene mayor relevancia puesto que todos están conectados a la red. A continuación, se enumeran y describen brevemente los elementos del hardware que conforman la instalación:

- Primera columna:
 - SCALANCE-XM416
 - SCALANCE-X208
 - Son dos modelos diferentes de switch. Son conmutadores de red, dispositivos de interconexión utilizados para conectar varios equipos en red.
 - Analizador de potencia del armario, para determinar las potencias y consumos eléctricos de la instalación.

- Segunda columna:
 - Conexión de las 5 ET200M, que corresponden a la periferia descentralizada de los elementos del armario.
- De la tercera columna en adelante:
 - Distribución de los 5 pupitres de la instalación y las cajas con las ET200S, que corresponden a la periferia descentralizada de los elementos de campo.

Nodo	IP	Dispositivo	Nombre	Comentario
2	172.22.121.1 172.22.121.2	Puerta de Enlace		
10	172.22.121.10	SW	MZI-1BS-E15-XM416	SCALANCE-XM416
11	172.22.121.11	CPU	MZI-1BS-E30-CPU	CPU 416-3F
12	172.22.121.12	CP443-1	MZI-1BS-E30-CP1	CP 443-1 (Ethernet + PROFINET)
13	172.22.121.13			
14	172.22.121.14			
15	172.22.121.15	HMI	MZI-1BS-E30-HM1	HMI TP1500 (+1BS)
16	172.22.121.16	HMI	MZI-1P01-E25-HM1	HMI TP900 (+1P01)
55	172.22.121.55	PAC3200	MZI-1SV-E10-AR1	SETRON PAC 3200
56	172.22.121.56	PN/PN	MZI-1BS-E15-PNP1	ACOPLADOR RED PROFINET
60	172.22.121.60	Switch	MZI-1BS-E15-SW1	SCALANCE-X208 (Anillo)
90	172.22.121.90	ET200M	MZI-1PS-E30-1M1	ET200M Modulo (+1PS)
91	172.22.121.91	ET200M	MZI-1AS1-E30-1M1	ET200M Modulo (+1AS1)
92	172.22.121.92	ET200M	MZI-1AS2-E30-1M1	ET200M Modulo (+1AS2)
93	172.22.121.93	ET200M	MZI-1AS3-E30-1M1	ET200M Modulo (+1AS3)
94	172.22.121.94	ET200M	MZI-1AS4-E30-1M1	ET200M Modulo (+1AS4)
95	172.22.121.95	ET200M	MZI-1AS5-E30-1M1	ET200M Modulo (+1AS5)
100	172.22.121.100	ET200S	MZI-C01.1-E25-1M1	ET200S Caja (+C01.1)
101	172.22.121.101	ET200S	MZI-C01.2-E25-1M1	ET200S Caja (+C01.2)
102	172.22.121.102	ET200S	MZI-C01.3-E25-1M1	ET200S Caja (+C01.3)
103	172.22.121.103	ET200S	MZI-C02.1-E25-1M1	ET200S Caja (+C02.1)
104	172.22.121.104	ET200S	MZI-C02.2-E25-1M1	ET200S Caja (+C02.2 DESAP.)
105	172.22.121.105	ET200S	MZI-C03.1-E25-1M1	ET200S Caja (+C03.1)
106	172.22.121.106	ET200S	MZI-C03.2-E25-1M1	ET200S Caja (+C03.2)
107	172.22.121.107	ET200S	MZI-C03.3-E25-1M1	ET200S Caja (+C03.3 APILA.)
108	172.22.121.108	ET200S	MZI-C04.1-E25-1M1	ET200S Caja (+C04.1)
109	172.22.121.109	ET200S	MZI-C04.2-E25-1M1	ET200S Caja (+C04.2)
110	172.22.121.110	ET200S	MZI-C04.3-E25-1M1	ET200S Caja (+C04.3)
111	172.22.121.111	ET200S	MZI-C05.1-E25-1M1	ET200S Caja (+C05.1)
120	172.22.121.120	ET200S	MZI-PL1-E25-1M1	ET200S PuertoLectura (+PL1)
130	172.22.121.130	Key Panel	MZI-P01-E25-KP1	Key panel KP32F (+P01)
131	172.22.121.131	Key Panel	MZI-P02-E25-KP1	Key panel KP32F (+P02)
132	172.22.121.132	Key Panel	MZI-P03-E25-KP1	Key panel KP32F (+P03)
133	172.22.121.133	Key Panel	MZI-P04-E25-KP1	Key panel KP32F (+P04)
134	172.22.121.134	Key Panel	MZI-P05-E25-KP1	Key panel KP32F (+P05)
150	172.22.121.150			ET200 Profisafe

Figura 17. Lista de elementos del hardware con direcciones IP de la instalación.

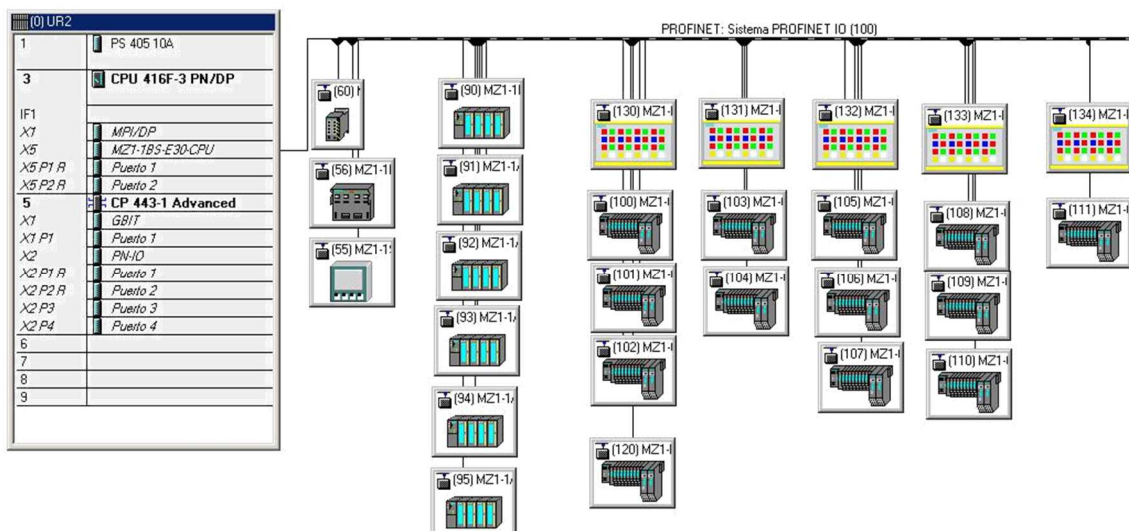


Figura 18. Ventana de configuración del hardware.

6.3 SÍMBOLOS DEL PROGRAMA

En todo programa de STEP 7, los símbolos son la nomenclatura utilizada para designar los elementos de programa que se utilizan para generar el código. Aquí es posible definir y modificar variables o símbolos, en lo que respecta a sus comentarios, funciones e incluso marcas. En la figura 19 se observa cómo acceder a la tabla de símbolos.

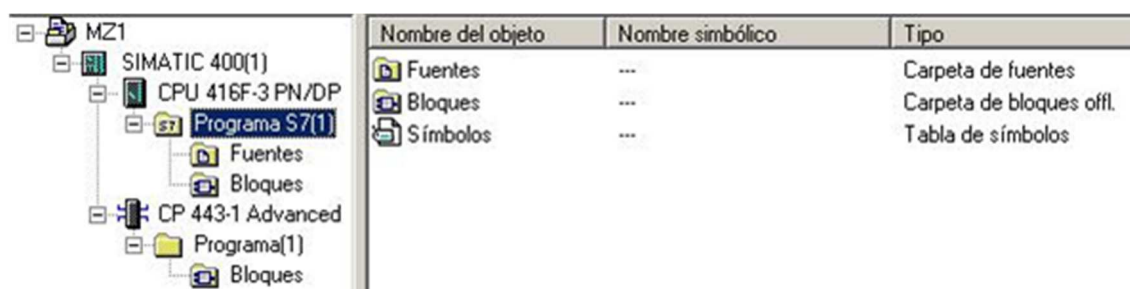


Figura 19. Ruta de acceso a la tabla de símbolos.

En la figura 20 se ha insertado un pequeño fragmento de la tabla de símbolos del proyecto como ejemplo, ya que no es posible mostrar aquí los casi 2000

símbolos de programa. Los datos que aparecen en la imagen son todos de tipo booleano, es decir, solo tienen dos posibles valores, 1 y 0. Estos símbolos hacen referencia a los detectores de presencia de carrocería. Para una cómoda programación, en el campo del símbolo, se coloca primero la mesa en cuestión y después el tipo de entrada con el objetivo de acceder rápidamente y de forma intuitiva a todas las entradas o salidas de cada mesa.

Estado	Símbolo	Dirección	Tipo de dato	Comentario
	+1.14.0_BE190	E 1030.0	BOOL	1.14.0 - BE190 Ocupado
	+1.14.0_BE191	E 1030.1	BOOL	1.14.0 - BE191 Cambio de Velocidad
	+1.14.1_BE110	E 1030.3	BOOL	1.14.1 - BE110 Ocupado 1
	+1.14.1_BE190	E 1030.2	BOOL	1.14.1 - BE190 Ocupado 2
	+1.14.2_BE190	E 1030.4	BOOL	1.14.2 - BE190 Ocupado
	+1.14.3_BE190	E 1030.5	BOOL	1.14.3 - BE190 Ocupado
	+1.14.4_BE190	E 1030.6	BOOL	1.14.4 - BE190 Ocupado
	+1.14.5_BE190	E 1060.0	BOOL	1.14.5 - BE190 Ocupado
	+1.14.6_BE190	E 1060.1	BOOL	1.14.6 - BE190 Ocupado
	+1.14.7_BE190	E 1060.2	BOOL	1.14.7 - BE190 Ocupado
	+1.14.8_BE190	E 1060.3	BOOL	1.14.8 - BE190 Ocupado
	+1.15.0_BE190	E 1030.7	BOOL	1.15.0 - BE190 Ocupado
	+1.15.0_BE191	E 1031.0	BOOL	1.15.0 - BE191 Cambio de Velocidad
	+1.15.1_BE110	E 1031.2	BOOL	1.15.1 - BE110 Ocupado 1
	+1.15.1_BE190	E 1031.1	BOOL	1.15.1 - BE190 Ocupado 2
	+1.15.2_BE190	E 1031.3	BOOL	1.15.2 - BE190 Ocupado
	+1.15.3_BE190	E 1031.4	BOOL	1.15.3 - BE190 Ocupado

Figura 20. Tabla de símbolos

6.4 BLOQUES DE FUNCIÓN OB

Para llevar a cabo un proyecto de estas dimensiones, es muy importante que el programa esté muy bien estructurado, es decir, es conveniente que la programación se organice en diversos bloques para poder acceder de forma cómoda a la parte del programa que se necesite en cada momento.

Por ese mismo motivo, en este proyecto, desde el bloque principal de programa OB1 simplemente se van a realizar las llamadas al resto de funciones del programa. Es decir, no se va a introducir la programación en este bloque. Por lo tanto, la función del OB1 es la “activación” del resto de funciones del programa.

6.5 BLOQUES DE FUNCIÓN FB

Llegados a este punto, ya se ha explicado en que entorno se ha desarrollado el proyecto en lo que respecta al software, Step 7 y TIA Portal, como programario universal de Siemens. A partir de estas explicaciones se ha descrito cómo se estructura el programa y los lenguajes que se han sido empleados. Además, también se ha puesto en contexto el hardware o los elementos principales de la instalación, como las redes y las configuraciones de los componentes a programar.

6.5.1 Descripción del problema

A lo largo de este apartado, se entra de lleno en el punto clave del desarrollo de código y del proyecto: la generación de un bloque único de programación que se pueda utilizar repetidas veces, ya que la lógica funcional de las mesas lo permite.

Es decir, se ha empleado la misma base de programación para todas las mesas, pero se ha adaptado para cada uno de los grupos funcionales. Estos grupos funcionales clasifican las distintas mesas con el siguiente criterio:

- Mesa de giro: son mesas que tienen un movimiento rotativo. Solo pueden transportar una única unidad de carrocería. Este tipo de mesa puede avanzar en los dos sentidos de giro, y cuenta con dos detectores de posición.
- Mesa elevadora: son mesas que tienen movimiento hacia posiciones arriba y abajo. Solo pueden transportar una única unidad de carrocería. Este tipo de mesa puede avanzar en los dos sentidos de elevación, y cuenta con dos detectores de posición, uno para cada sentido de avance.
- Mesa de rodillos: Aunque las mesas de giro y elevadoras también tienen rodillos, estas son mesas que solo tienen la función de mover rodillos hacia adelante o hacia atrás en el sentido de la marcha de la carrocería.

En este grupo funcional es donde se encuentra más variedad en los tipos según el número de carrocerías que caben, detectores de posición y sentidos. A continuación, se presentan los distintos tipos:

- Mesa de rodillos de 2 carrocerías, 1 detector de posición y 1 sentido.
- Mesa de rodillos de 2 carrocerías, 2 detectores de posición y 1 sentido.
- Mesa de rodillos de 1 carrocería, 1 detectores de posición y 1 sentido.
- Mesa de rodillos de 1 carrocería, 2 detectores de posición y 2 sentidos.

En definitiva, si en la instalación hay 40 mesas, de las cuales 30 son de rodillos, 5 son de giro y 5 son elevadoras, con únicamente tres bloques funcionales se consigue tener la programación de toda la instalación, con tan solo cambiar las condiciones de entrada y de salida según el tipo de mesa. No es necesario hacer un FB por cada mesa con sus respectivos sensores y motores a activar. Así se consigue ahorrar muchos recursos. En la figura 21 se observa cómo se visualiza un FB en SIMATIC.

Seguidamente se explicará un FB de una mesa de rodillos de 1 carrocería, 2 detectores y 2 sentidos. Una vez explicado la lógica de este bloque, adaptándola con cambios mínimos quedarían establecidos el resto de los bloques de mesas de rodillos, de giro y elevadoras.



Nombre del objeto	Nombre simbólico	Lenguaje	Tamaño en la memor...	Tipo	Versión (en)
Datos de sistema	---	---	---	SDB	---
OB1	---	KOP	48	Bloque de organizaci...	0.1
FB70	MR_2D_2S	KOP	1052	Bloque de función	0.1
FB71	MR_2P_2D_1S	KOP	970	Bloque de función	0.1
FB72	MR_1D_1S	KOP	614	Bloque de función	0.1

Figura 21. Representación de un bloque FB en el entorno SIMATIC

6.5.2 Metodología empleada

En este apartado se describirá de una manera sencilla, cuál ha sido la metodología en la cual se ha basado el desarrollo de las funciones modelo, que son las FB.

Para empezar, se describen todos los estados (Step Cycles) en los cuales se puede encontrar una mesa:

- 00: Inicio de ciclo
- 01: Mesa vacía
- 02: Tránsito de carga directo
- 03: Tránsito de carga inverso
- 04: Mesa ocupada
- 05: Tránsito de descarga directo
- 06: Tránsito de descarga inverso

Existen tres etapas en la secuencia de transporte de la carrocería entre una mesa y la siguiente. En todas las etapas es condición imprescindible que la instalación se encuentre operando en automático y sin fallos. Las etapas se describen a continuación:

- Etapa de autorización de carga: la autorización de carga es una señal que se envía desde la mesa posterior a la mesa anterior. Para que se genere esa señal, es necesario que la mesa posterior esté vacía o en tránsito de carga, y que no se encuentre en ninguno de los otros estados (en tránsito de descarga o en carga en el otro sentido). Esta etapa se muestra en la figura 22.

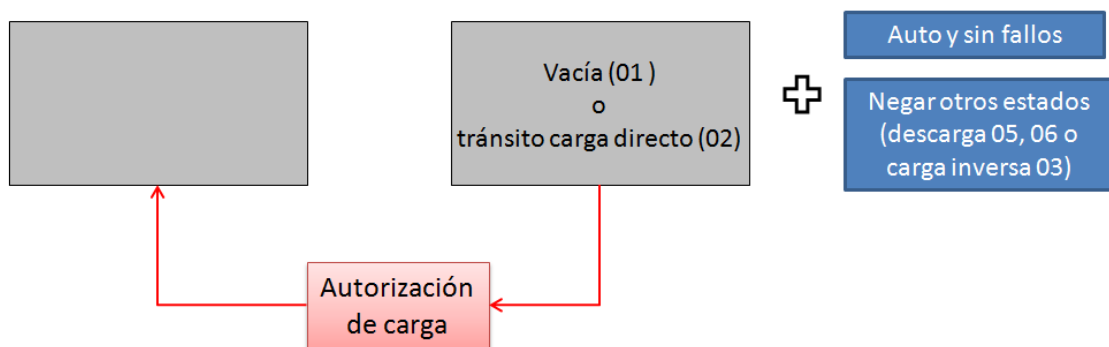


Figura 22. Representación de la etapa de autorización de carga entre mesas en sentido directo

- Etapa de petición de descarga: la petición de descarga es una señal que se envía desde la mesa anterior a la posterior. Para que se genere esa señal, es necesario que la mesa anterior esté ocupada o en tránsito de descarga. Esta etapa se muestra en la figura 23.

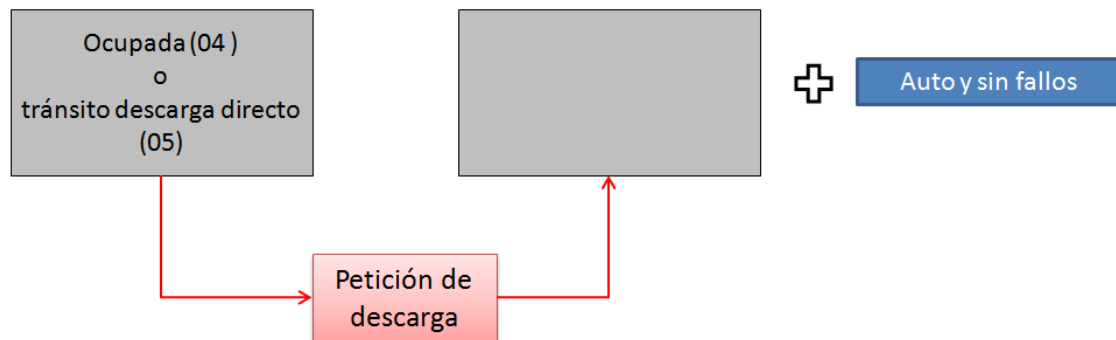


Figura 23. Representación de la etapa de petición de descarga entre mesas en sentido directo

- Etapa de fin de tránsito: el fin de tránsito es una señal que se envía desde la mesa posterior a la anterior. Para que se genere esa señal, es necesario que la mesa posterior esté ocupada y con el detector de presencia activo. Esta etapa se muestra en la figura 24.

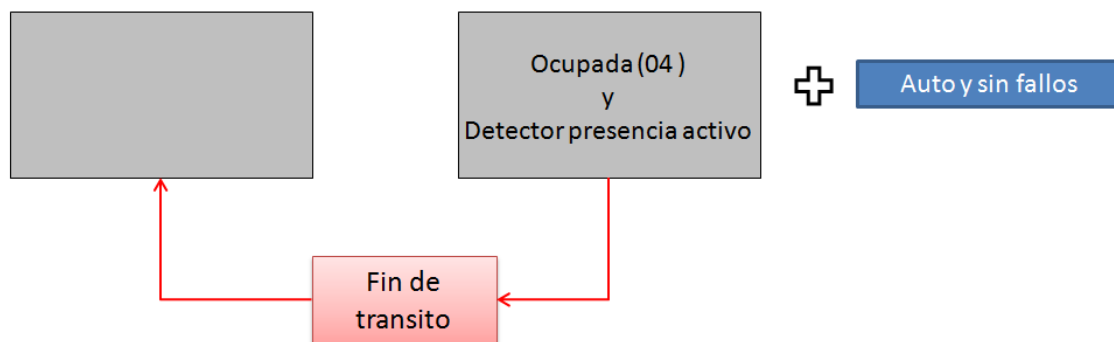


Figura 24. Representación de fin de tránsito.

Para mayor claridad, solo se ha explicado la secuencia en sentido directo. Si fuera en sentido inverso, las etapas serían iguales, pero activando los estados de tránsito en sentido inverso (estados 03 y 06)

6.5.3 Secuencia de programación

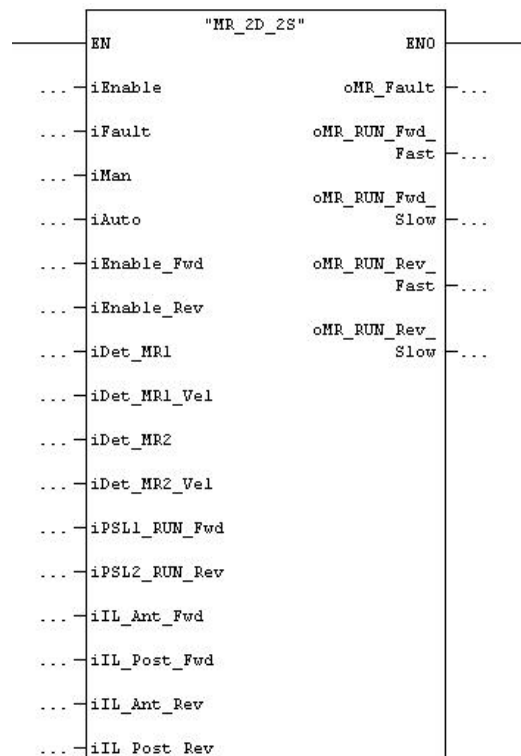


Figura 25. Función FB MR_2D_2S

En la figura 25 se encuentra la llamada al bloque FB MR_2D_2S, el cual pertenece a una mesa de rodillos de 1 carrocería, 2 detectores y 2 sentidos. Dicho cuadro dispone de una serie de entradas en su parte izquierda, referenciadas a unas entradas que hemos dispuesto dentro del mismo. En la parte derecha se encuentran las salidas, las cuales se utilizan del mismo modo. En cuanto a las entradas, “iEnable” activa el bloque, iFault corresponde a si la mesa se encuentra en fallo, iMan hace referencia al modo manual mientras que iAuto lo hace al modo automático y las entradas iDet corresponden a los detectores MR1 y MR2, tanto de posición como de velocidad. Además, se dispone de los botones de RUN en sentido directo e inverso, iIL_Ant_Fwd comunica la mesa anterior en sentido directo con la mesa a la que corresponda este bloque e iIL_Post_Fwd le comunica con la siguiente también en sentido directo. Por otro lado, en iIL_Ant_Rev se coloca la dirección de la mesa que en sentido normal sería la siguiente, pero en caso inverso es anterior. Por último,

ilL_Post_Rev albergará la dirección de la mesa siguiente en sentido inverso. En cuanto a las salidas, se dispone del aviso de fallo de mesa y de la activación del motor rápido o lento en sentido directo e inverso.

Adentrándose en el interior del bloque para ver la programación que se ha realizado, el primer paso es realizar un reinicio de todas las estructuras temporales para asegurar que se encuentran a cero. Además, también al principio de la función, se genera una marca que se va a encontrar siempre a 1 y otra que se encuentre siempre a 0, de modo que sea posible utilizarla siempre que se requiera una entrada con uno de esos valores.

Tras el primer segmento de la figura 26, para generar la marca siempre a 1, como se observa en el segmento 2, se coloca en paralelo un contacto negado y otro sin negar referentes a la misma marca, por lo que la salida siempre estará conectada. En cuanto a la señal siempre a cero, se colocan otros dos contactos (uno normalmente abierto y uno normalmente cerrado) referentes a la misma marca, pero esta vez en serie, por lo que la salida nunca llegará a recibir un 1.

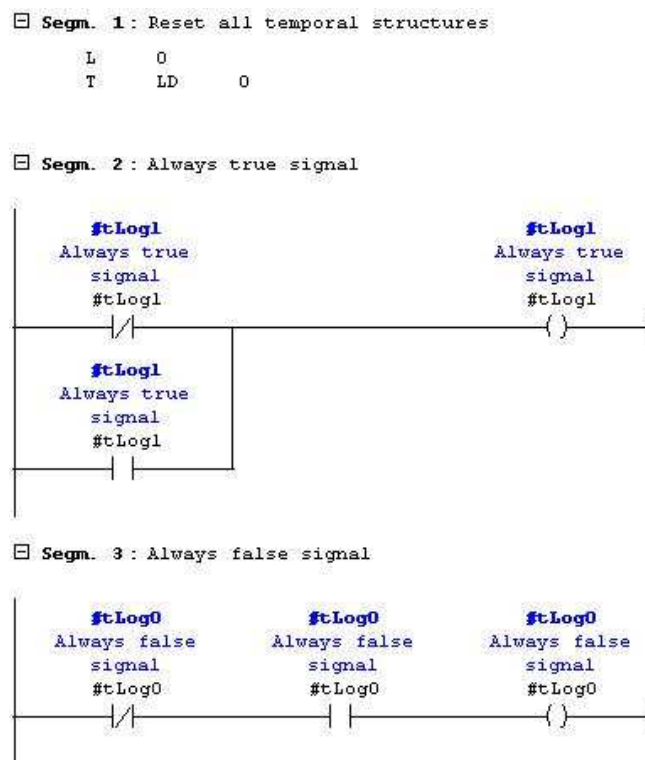


Figura 26. Reset temporales, señal siempre a 0 y a 1

Tras estos primeros segmentos, se pasa a la programación específica del bloque. En el siguiente segmento se ha llevado a cabo las entradas de tránsitos en modo directo. Tras una marca que siempre se encuentra a 1, se dividen tres ramas en paralelo como se observa en la figura 27. En la primera de estas, se programa la petición de descarga que se recibe de la mesa anterior, en la siguiente rama se dispone de la autorización de carga que se entrega a la mesa anterior y, por último, el fin de tránsito que también se entrega a la mesa posterior.

La petición de descarga de la mesa anterior se activa cuando dicha mesa contiene una carrocería en correcta posición y no se encuentra en fallo. La autorización de carga se genera cuando se recibe una petición de descarga mientras que no se contiene ninguna carrocería en la mesa. Por último, cuando se ha recibido la carrocería sin ningún tipo de problema, se genera una señal de fin de tránsito.

Segm. 4 : Entradas transitos directa

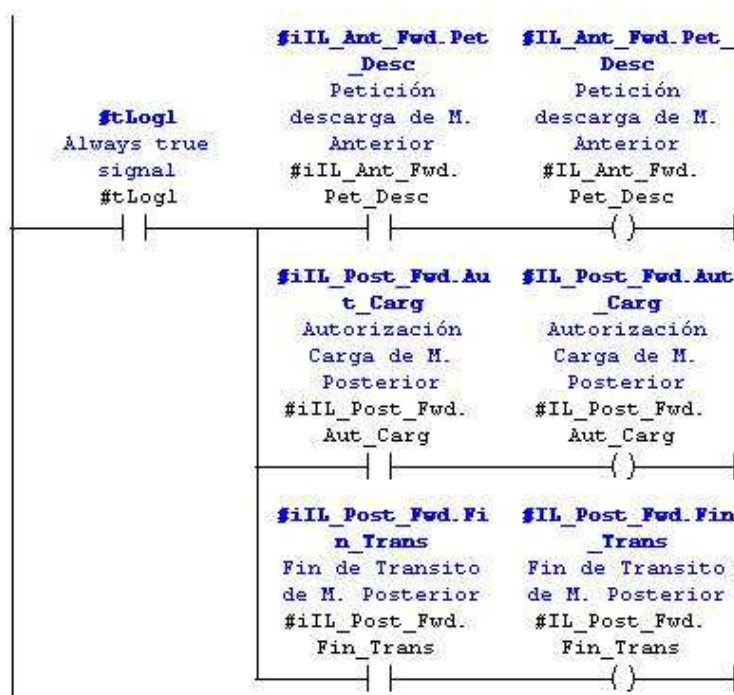


Figura 27. Entradas de transitos de forma directa

Tras este segmento, se realiza la programación de otro con la misma estructura, pero esta vez con las señales invertidas, de forma que se corresponde a las entradas de los transitos en sentido inverso.

Se continua con la programación de los posibles defectos de presencias en los detectores. Como se aprecia en la figura 28, cuando se encuentre en el estado Step Cycle 01 correspondiente a la mesa vacía y, al mismo tiempo, se encuentren a 1 alguno de los detectores de presencia o de cambio de velocidad, se activará la señal de defecto en detectores. Además, si la mesa se encuentra ocupada en Step Cycle 04 y ninguno de los detectores de presencia se encuentra a 1, también se activará la señal de salida. Los Step Cycles son los estados en que se encuentra la mesa.

Segm. 6 : DEFECTO PRESENCIAS

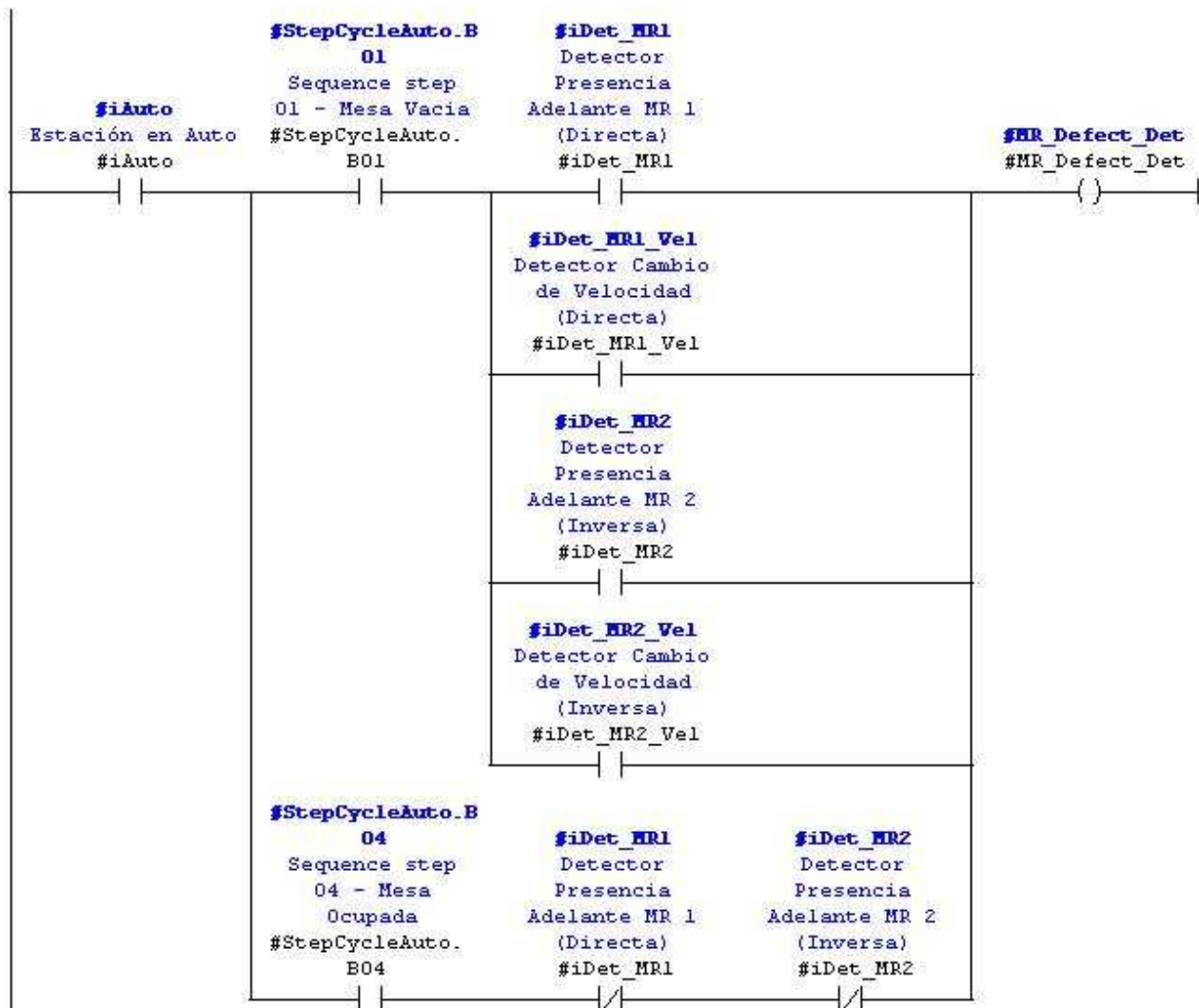


Figura 28. Defecto de presencias

Se prosigue con el resumen de fallos de la mesa, es decir, una salida que se active cuando se accione cualquiera de los fallos de la mesa, así como los defectos de sus detectores o un posible reset de emergencias. Este segmento se muestra en la figura 29.

Segm. 7 : tFault: Resumen de fallos

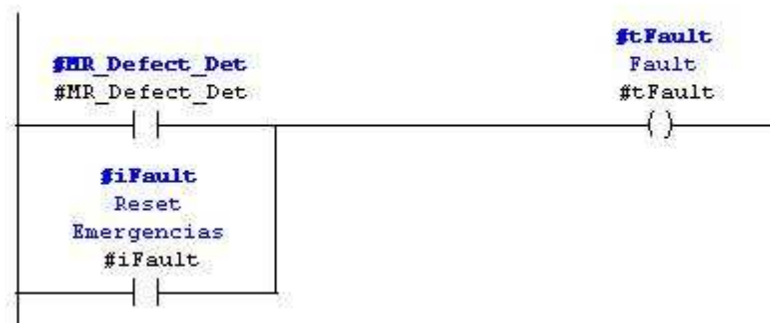


Figura 29. Resumen de fallos

Tras el resumen de fallos, se sigue con el inicio de la secuencia como se contempla en la figura 30. Cuando la mesa no se encuentre en fallo y, además, no esté en ninguno de los estados posibles, se activa la señal de inicio de secuencia Init_seq.

Segm. 8 : Iniciar seq.



Figura 30. Inicio de secuencia

Para la correcta programación de la mesa, es importante asegurarse de conocer en todo momento si la mesa se encuentra en algún tipo de tránsito. Para ello, se genera una señal correspondiente a ningún tránsito directo que se active cuando la mesa no se encuentre en los tránsitos de carga o descarga directa. Además, se dispone de una bobina que se activa cuando no se encuentran activos los Step Cycles 03 y 06, es decir, los tránsitos de carga y descarga inversa. Se puede distinguir en la figura 31.

Segm. 9 : NINGUN TRANSITO DIRECTA



Segm. 10 : NINGUN TRANSITO INVERSA



Figura 31. Tránsitos de carga y descarga

En el siguiente segmento, en la figura 32, se lleva a cabo el reseteo de los tránsitos cuando se mueve la mesa en modo manual. Si se observan los contactos, tras la activación de la estación en estado manual y el pulsador de marcha de los rodillos en sentido directo o inverso, se realiza un reinicio de los cuatro tránsitos correspondientes a la carga y descarga en ambos sentidos.

Segm. 11 : RESET TRANSITOS

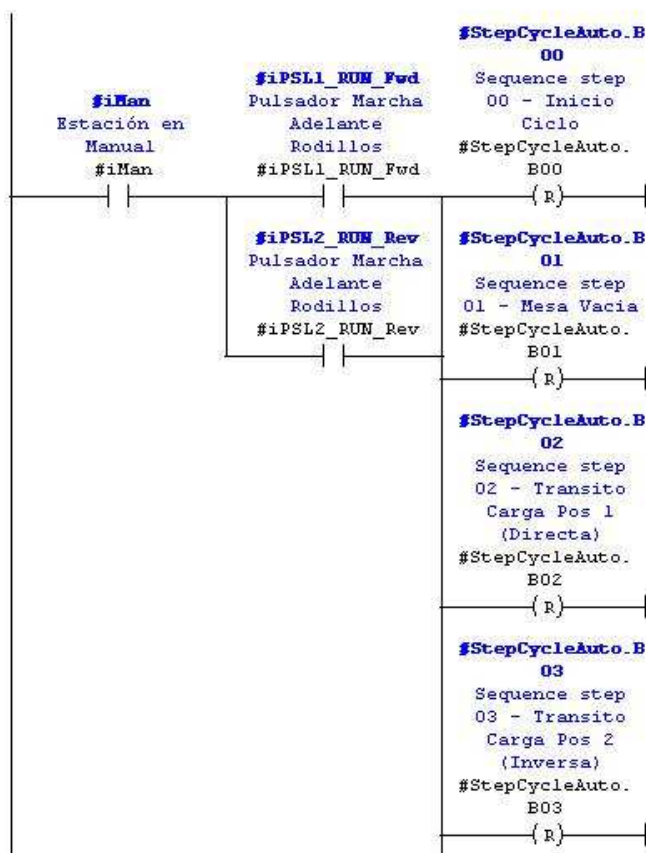


Figura 32. Reinicio de los tránsitos

En la figura 33, se lleva a cabo el inicio del ciclo. Cuando la estación se encuentra en modo automático, sin errores y con el inicio de secuencia activo, se activa el Step Cycle 00.

Segm. 12 : Sequence step 00 - Cycle start request

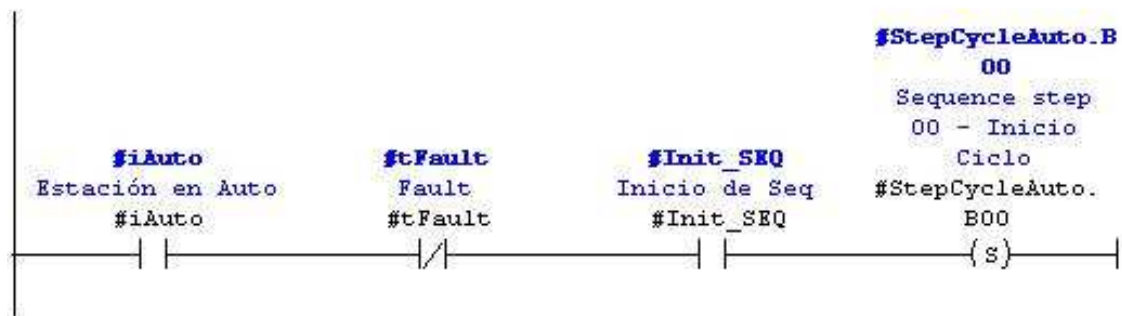


Figura 33. Activación Step Cycle 00

Se prosigue con el ciclo 00 como tal, el cual, al igual que el resto, necesita que la estación se encuentre en automático y sin errores, por lo que son los primeros contactos que se colocan en serie. Tras esto, con el ciclo todavía en activo y los dos detectores sin detectar carrocería, se activa el Step Cycle 01 correspondiente a mesa vacía y se reinicia el estado actual. Se observa en la figura 34.

Segm. 13: Sequence step 00 - Cycle start request

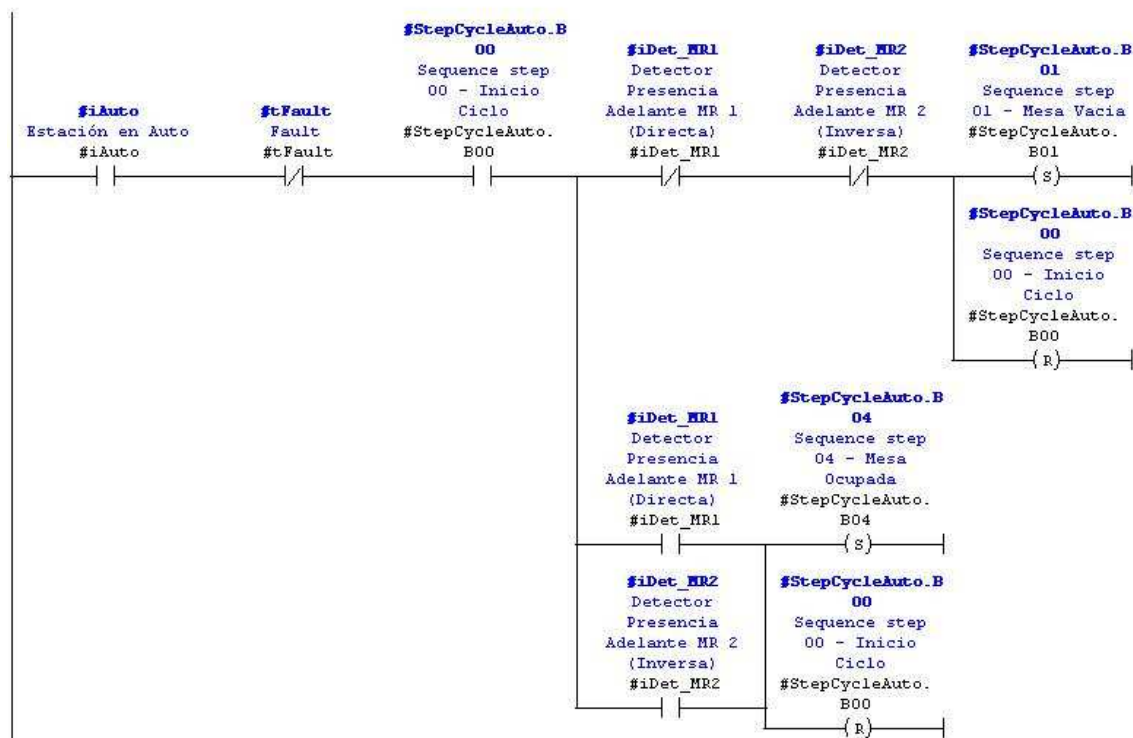


Figura 34. Sequence step 00

No obstante, tal y como se aprecia en la figura 34, si alguno de los dos detectores se activa debido a la captación de una carrocería, se pasa al Step Cycle 04, correspondiente a mesa ocupada. Del mismo modo que en el anterior caso, también se reinicia el estado actual (Step Cycle 00).

Al igual que en el ciclo anterior, se sitúan los primeros contactos de estación en modo automático y mesa sin errores, así como un contacto normalmente abierto correspondiente al estado actual, el Step Cycle 01. Cuando estas condiciones se cumplen, junto con la petición de descarga de la mesa anterior, se pasa al Step Cycle 02, tránsito de carga directa, reseteando el estado actual en la figura 35.

Sin embargo, si la petición de carga de la mesa anterior corresponde a la que sería la siguiente en estado directo, se continua al Step Cycle 03, correspondiente al tránsito de carga inverso. Al mismo tiempo, se realiza un reinicio del ciclo anterior.

Segm. 14 : Sequence step 01 - Mesa Vacia Pos 1, Pos 2.

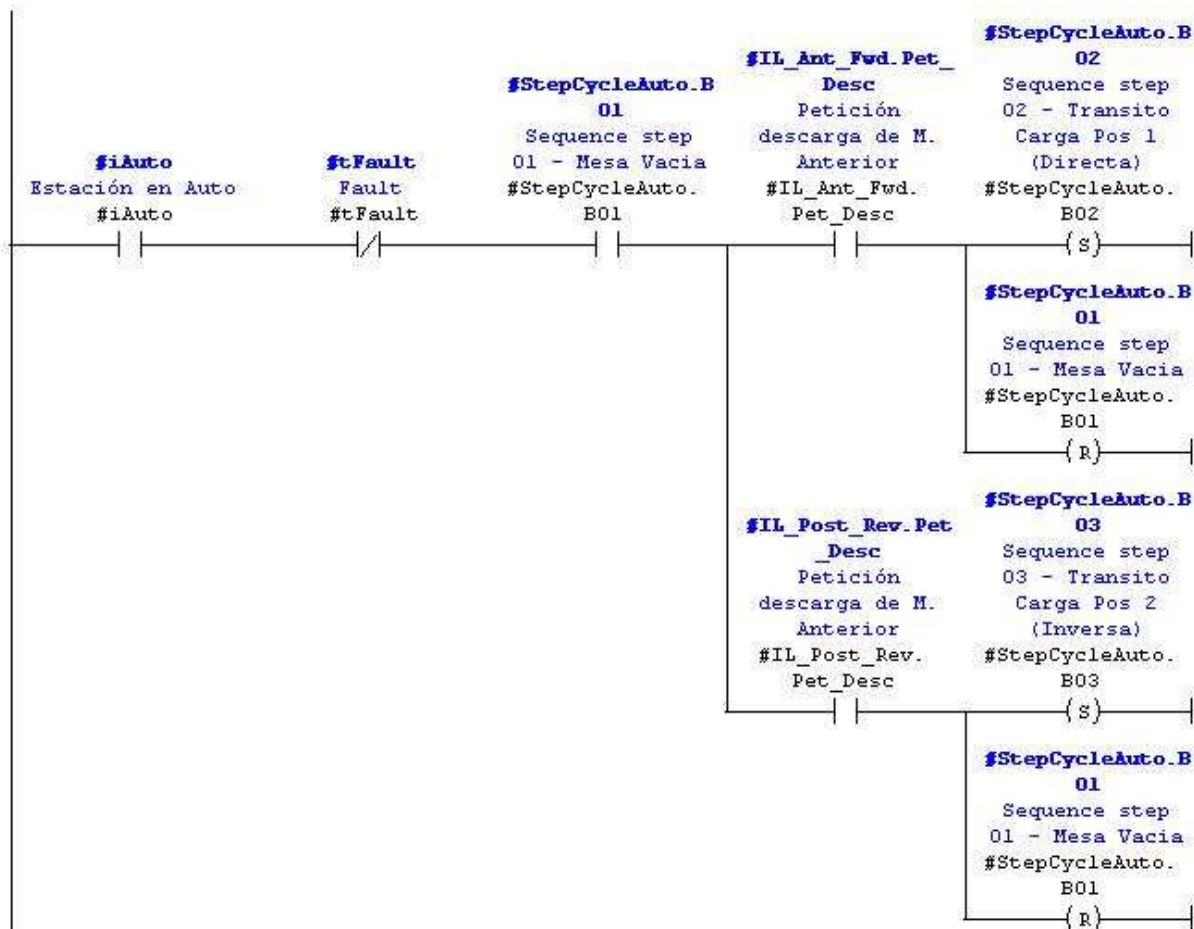


Figura 35. Sequence Step 01

Segm. 15 : Sequence step 02 - Transito Carga Pos 1



Segm. 16 : Sequence step 02 - Transito Carga Pos 2 (Inversa)



Figura 36. Sequence Step 02 y 03

Se continua con las secuencias 02 y 03 en la figura 36. La primera de estas se reiniciará cuando se cumplan las condiciones previas que son que el propio ciclo se encuentre activo junto con el detector MR1 y, además, lo esté la estación en automático mientras no lo esté la mesa. En cuanto a la secuencia 03, se reiniciará del mismo modo que la anterior sustituyendo el detector MR1 por el MR2 y el ciclo correspondiente.

El siguiente segmento alberga el Step Cycle correspondiente a la situación en que la mesa se encuentra ocupada. Aquí, al igual que en los casos anteriores, cuando se encuentre la estación en automático y sin fallos, se pasa a comprobar que el Step Cycle 04 está activo. Tras esto, si la mesa posterior activa la autorización de carga, en función del sentido de la marcha, se activará el Step Cycle 05 o el Step Cycle 06, reiniciando al mismo tiempo el estado correspondiente a "Mesa Ocupada". Se aprecia en la figura 37.

Segm. 17 : Sequence step 03 - Mesa Ocupada

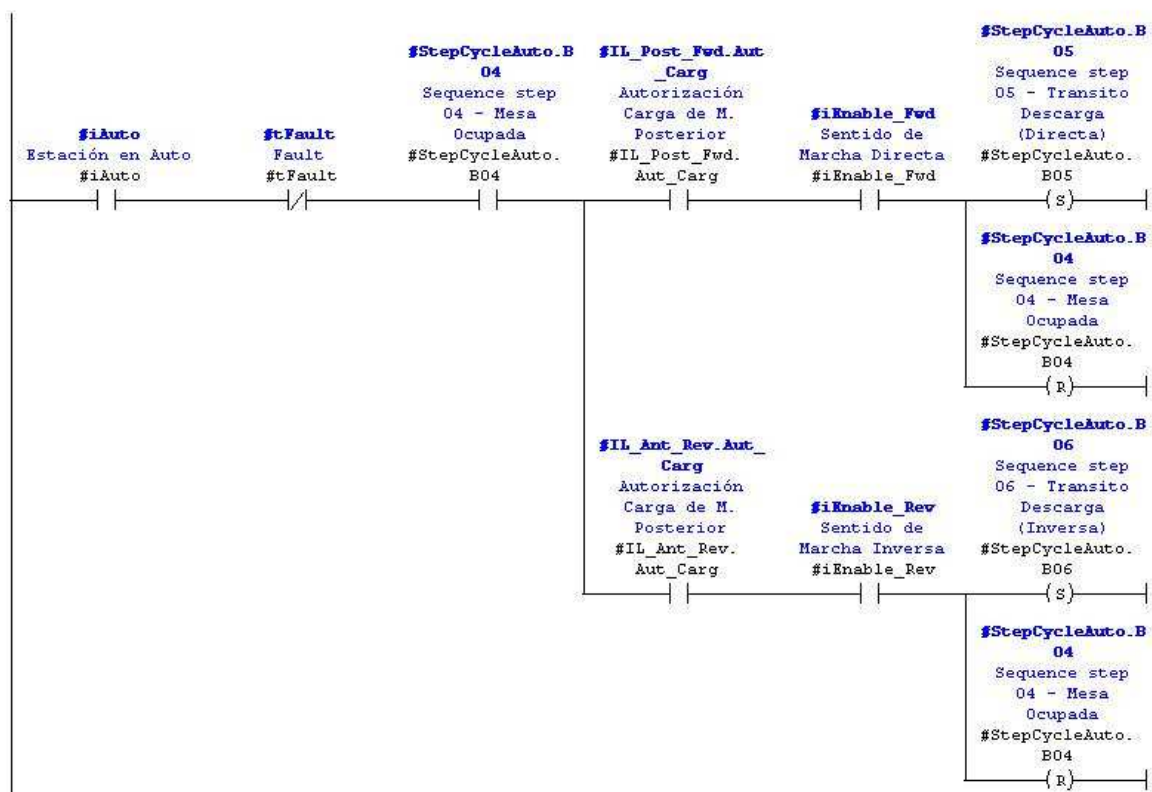


Figura 37. Sequence Step 04

Se pasa a programar los tránsitos de descarga, tanto directa como inversa, correspondientes a los Step Cycles 05 y 06 en la figura 38. Tras la comprobación de la estación en automático y que no se encuentre en fallo, se comprueba que está activo el tránsito correspondiente junto con la señal de fin de tránsito de la mesa posterior para finalizar dicho ciclo reiniciándolo al final del segmento.

Segm. 18: Sequence step 04 - Transito Descarga (Directa)



Segm. 19: Sequence step 04 - Transito Descarga (Inversa)



Figura 38. Sequence Step 05 y 06

Se continúa ahora con las autorizaciones, tanto de carga como de descarga y fines de tránsitos en la figura 39. Se comienza con las autorizaciones de carga de la mesa de rodillos de forma directa.

Segm. 23: Autorización Carga mesa Rodillos (Directa)

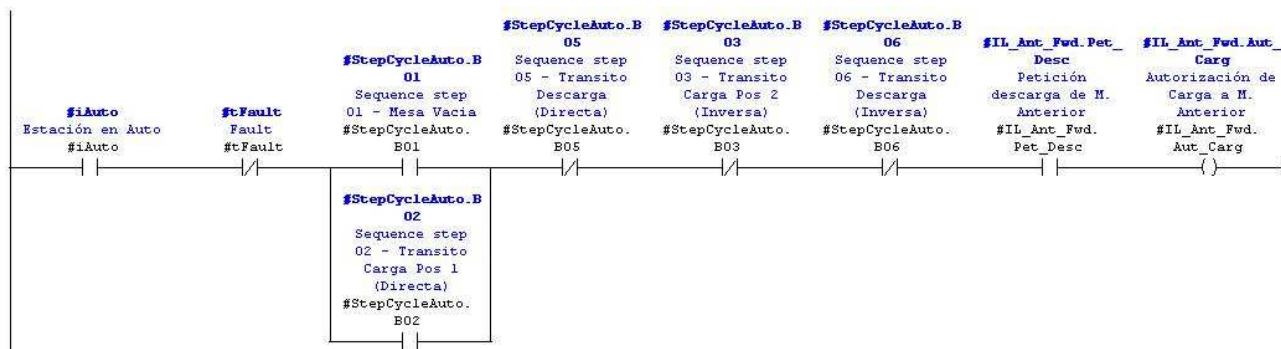


Figura 39. Autorización carga directa

Tras la comprobación de la estación en automático y sin fallo, mientras la mesa se encuentre vacía o en tránsito de carga directa, sin ningún otro tipo de tránsito y con la petición de descarga de la mesa anterior, se activa la autorización de carga.

El siguiente segmento, en la figura 40, corresponde a la activación de la autorización de carga de la mesa de rodillos en sentido inverso. En este caso, la diferencia es que el cambio del contacto correspondiente al tránsito de carga directa por el de tránsito de carga inversa. Además, también se sustituye la petición de descarga de mesa anterior en sentido directo por la misma de sentido inverso.

Segm. 24 : Autorización Carga mesa Rodillos (Inversa)

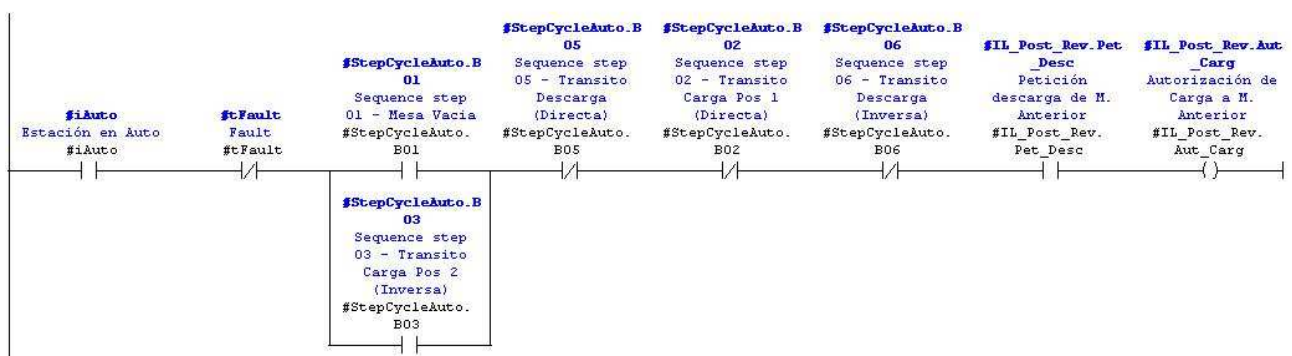


Figura 40. Autorización de carga inversa

Tras las autorizaciones de carga, en la figura 41, se programan las peticiones de descarga a la mesa posterior. Se empieza con la petición de descarga a la mesa posterior en sentido directo, la cual se activa cuando se cumplan las condiciones previas que se aprecian en la figura 41, tales como el sentido de marcha directo, la habilitación, la estación en estado automático, que la mesa no se encuentre en error y uno de los Step Cycles correspondientes a mesa ocupada o tránsito de descarga directa.

Segm. 25 : Peticion Descarga Mesa de Rodillos

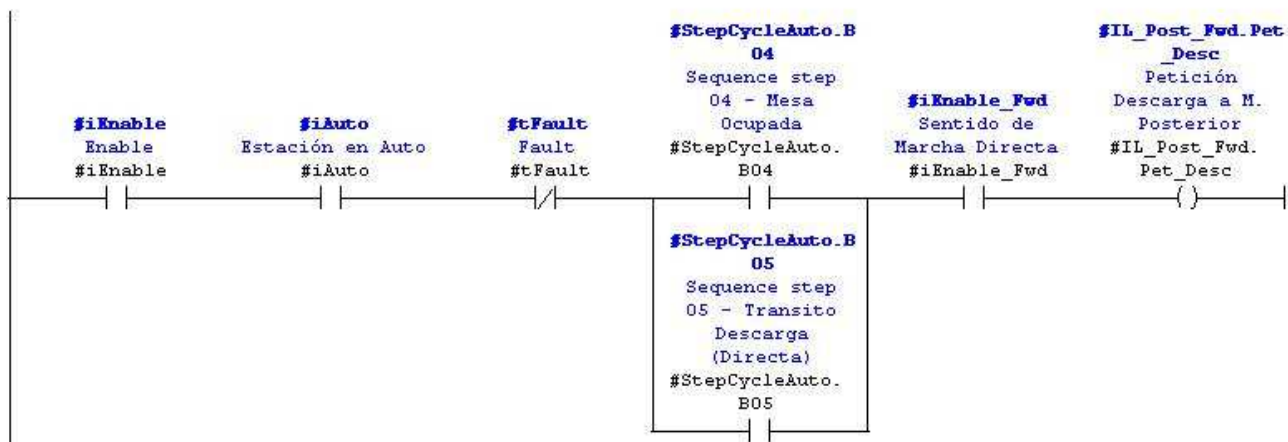


Figura 41. Petición de descarga en sentido directo

En cuanto a la petición de descarga a la mesa posterior en sentido inverso, se activa cuando se cumplan las mismas condiciones que en el segmento anterior, sustituyendo los contactos correspondientes a sentido directo por sentido inverso. Se puede apreciar en la figura 42.

Segm. 26 : Peticion Descarga Mesa de Rodillos

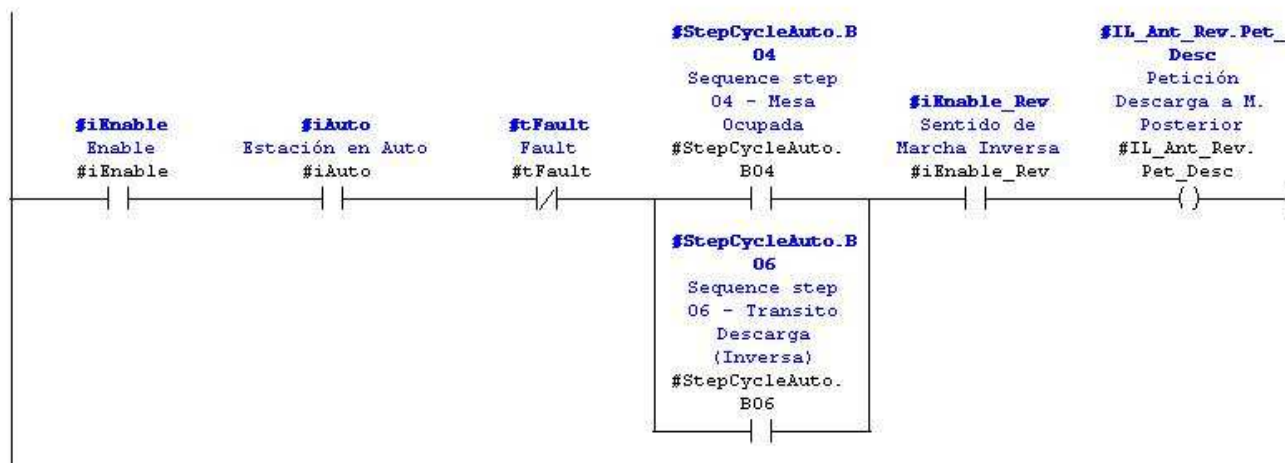


Figura 42. Petición de descarga en sentido inverso

Se sigue con las señales de fin de tránsito, las cuales, junto con la estación en automático y sin fallos, además de la mesa ocupada pasarán a activar el fin de

tránsito a mesa anterior directo o inverso en función de si se ha activado el detector MR1 o el MR2, además del pulso Fin_Fwd o Fin_Rev. Se puede ver en la figura 43.

Segm. 27 : Fin Transito Rodillos (Directa)



Segm. 28 : Fin Transito Rodillos (Directa)

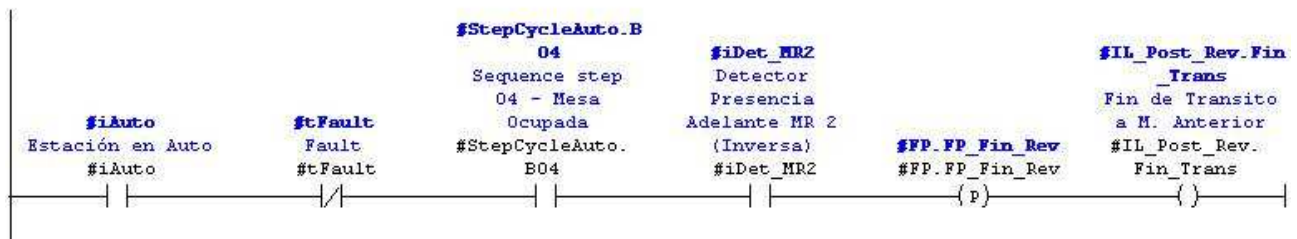
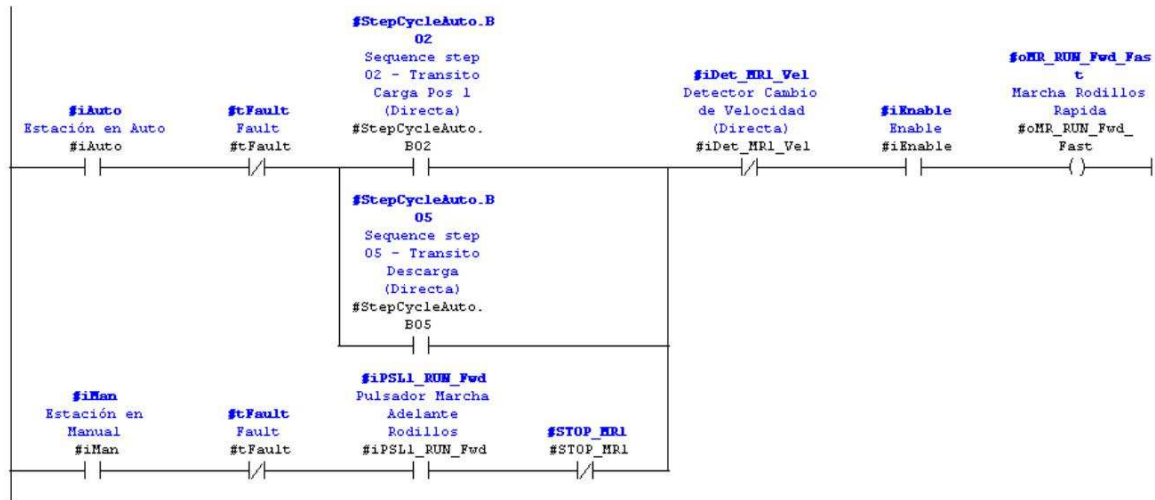


Figura 43. Fin de tránsito directo e inverso

Continuando con los últimos segmentos, se lleva a cabo la programación de la marcha rápida y la marcha lenta de rodillos. La programación, tal y como se aprecia en la figura 44, se ha realizado del mismo modo cambiando únicamente el contacto del detector de cambio de velocidad, que se alterna mediante un selector del pupitre.

Segm. 29: OUT RUN RODILLOS RAPIDA (Directa)



Segm. 30: OUT RUN RODILLOS LENTA (Directa)

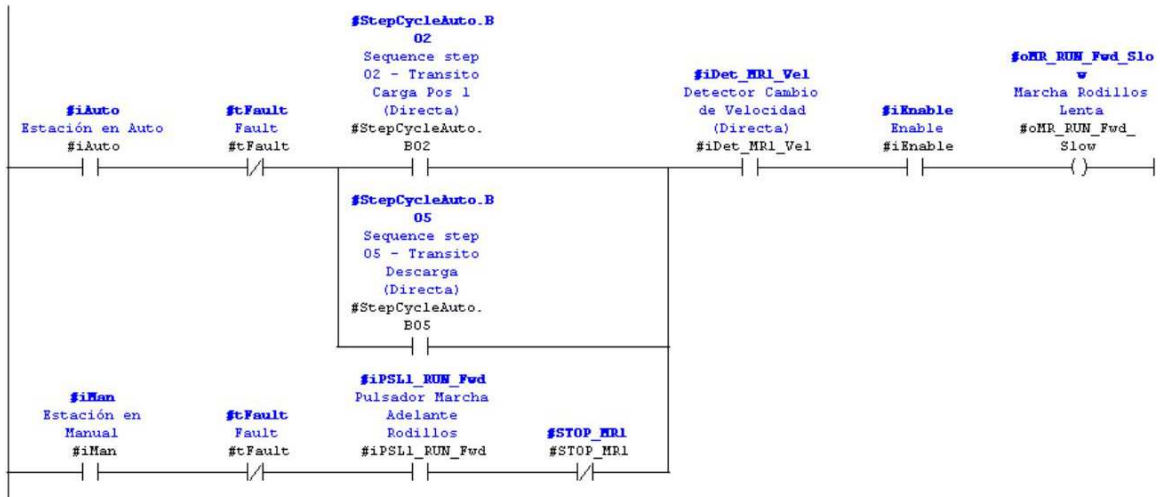


Figura 44. Salidas marcha rodillos rápida y lenta

Por último, en la figura 45, se ha generado una salida que avise cuando la mesa de rodillos se encuentra en fallo, activada por la variable `#tFault`, la cual se activa con cualquier tipo de fallo en la mesa.

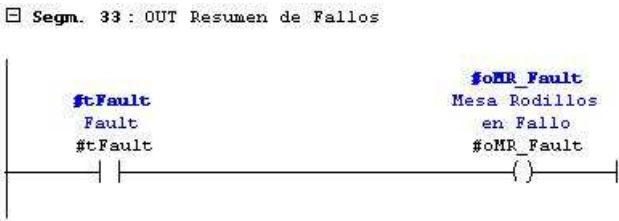


Figura 45. Salida de resumen de fallos

6.6 BLOQUES DE FUNCIÓN FC

Primeramente, se tiene que decir que hay tantas FC como mesas dispone la instalación. Se puede observar cómo se visualizan en el SIMATIC en la figura 46. Las FC son las funciones donde se llama a la FB correspondiente según el grupo funcional al que pertenezca la mesa. Es decir, habrán FCs con la misma FB. En cada FC se programan las averías de la mesa que vienen de las salidas de la FB que se llama dentro. Además, puede haber averías programadas de protecciones de los motores de la mesa. También se han empleado para generar las averías principales del armario y del hardware en general.

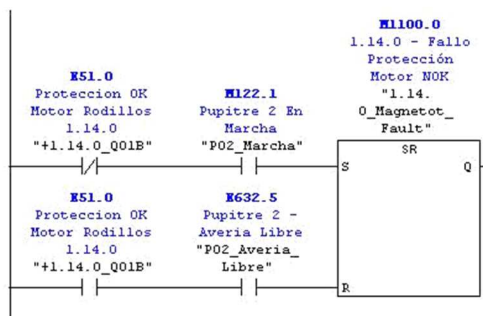
FC100	SG1: FALLO HARDWARE	KOP	670	Función
FC110	SG1: [P01] PUP 1 FALLO	KOP	132	Función
FC111	SG1: [P02] PUP 2 FALLO	KOP	96	Función
FC112	SG1: [P03] PUP 3 FALLO	KOP	114	Función
FC113	SG1: [P04] PUP 4 FALLO	KOP	114	Función
FC114	SG1: [P05] PUP 5 FALLO	KOP	78	Función
FC1150	1.14.5	KOP	816	
FC1160	1.14.6	KOP	816	
FC1170	1.14.7	KOP	816	
FC1180	1.14.8	KOP	868	
FC1190	1.15.0	KOP	798	
FC1200	1.15.1	KOP	856	
FC1210	1.15.2	KOP	816	
FC1220	1.15.3	KOP	816	

Figura 46. Bloques de función FC representados en SIMATIC.

Se utilizan FCs en vez de FBs porque, como se ha comentado, los FC no tienen memoria propia y no se generan DBs innecesarios al utilizarlos. En cambio, para los grupos funcionales de mesa es más eficiente trabajar con su propia memoria a tener que generar DBs a parte.

En la figura 47 se observa parte del interior de una FC de una mesa donde están programados los fallos de protección del motor y el fallo de defecto de tiempo. Están programadas con marcas que servirán para visualizar en la pantalla y son de fácil acceso. A continuación, en la figura 48, se observa otra parte del interior de una FC de una mesa en el que se llama a la FB del bloque funcional correspondiente y se le aplican las entradas/salidas que toca.

Segm. 1: Fallo Proteccion



Segm. 2: Fallo Defecto Tiempo

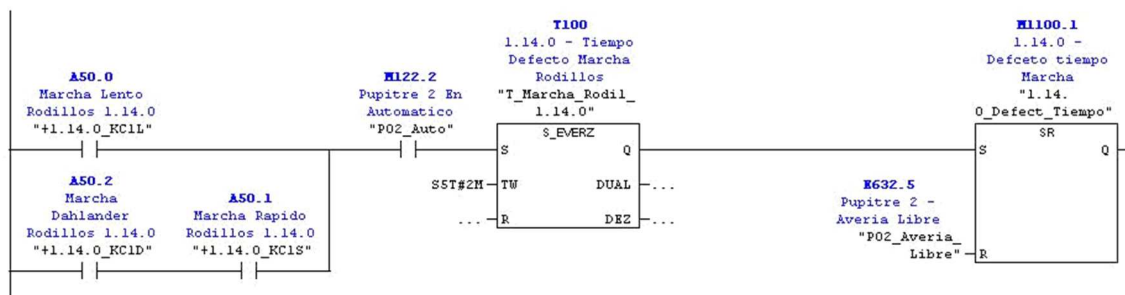


Figura 47. Fallos en el FC de una mesa

Segm. 10 : Titulo:

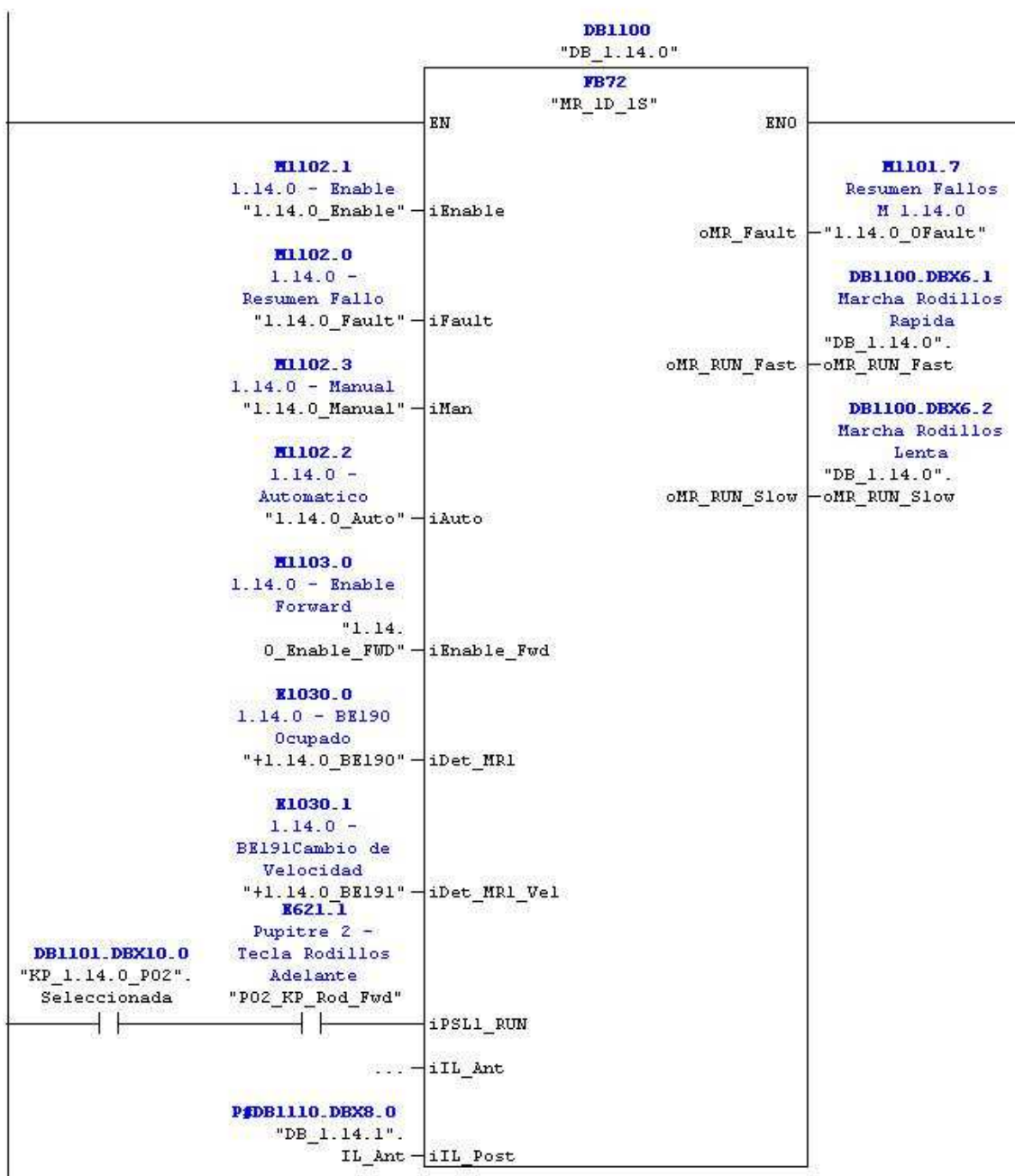


Figura 48. Llamada a una FB dentro de una FC de mesa

6.7 BLOQUES DE DATOS DB

Se ha tenido que generar un bloque de datos para cada mesa de la instalación. Este DB es a su vez DB de instancia del FB del grupo funcional del que sea dicha mesa. Esto ha sido necesario porque cuando se utiliza la misma función en distintas mesas, se corre el riesgo de solapar datos si se utiliza el mismo DB. En la figura 49 se puede observar una parte de la lista de DBs del programa.

DB1150	DB_1.14.5	DB	54	DB de instancia del FB 74
DB1151	KP_1.14.5_P03	DB	48	DB de instancia del FB 95
DB1160	DB_1.14.6	DB	54	DB de instancia del FB 74
DB1161	KP_1.14.6_P03	DB	48	DB de instancia del FB 95
DB1170	DB_1.14.7	DB	54	DB de instancia del FB 74
DB1171	KP_1.14.7_P03	DB	48	DB de instancia del FB 95
DB1180	DB_1.14.8	DB	54	DB de instancia del FB 73
DB1181	KP_1.14.8_P03	DB	48	DB de instancia del FB 95
DB1190	DB_1.15.0	DB	54	DB de instancia del FB 72
DB1191	KP_1.15.0_P02	DB	48	DB de instancia del FB 95
DB1200	DB_1.15.1	DB	54	DB de instancia del FB 71
DB1201	KP_1.15.1_P02	DB	48	DB de instancia del FB 95
DB1210	DB_1.15.2	DB	54	DB de instancia del FB 74

Figura 49. Lista de DB del programa en el entorno de SIMATIC

En el interior de estos bloques se encuentran las variables correspondientes a la función sobre la que se ha generado junto con su dirección, su tipo de dato, su valor inicial asignado, el valor actual si se encuentra en modo de observación y un comentario acerca de la variable. En la figura 50, se muestra un fragmento de un DB del programa a modo de ejemplo.

Dirección	Nombre	Tipo	Valor inic.	Comentario
0.0		STRUCT		
+0.0	NUMERO	INT	0	
+2.0	FLAG	STRUCT		
+0.0	Identificado	BOOL	FALSE	Skid Identificado (en proceso)
+0.1	Carroceria	BOOL	FALSE	Skid con carroceria
+0.2	Pieza	BOOL	FALSE	Skid con pieza

Figura 50. Fragmento del bloque de datos de una FB

6.8 BLOQUES DE SEGURIDAD

Finalmente, también están los bloques del sistema de seguridad. La configuración del sistema de seguridad es idéntica al de automatización estándar. Se programan en KOP y AWL. La distinción en el entorno de SIMATIC es que estos bloques aparecen en color amarillo, como muestra la figura 51.

La principal función de estos bloques es la configuración para intervenir en la instalación en situaciones en las que se requiere un elemento de emergencia como una barrera de seguridad, una seta de emergencia, etc.








	FC5000	SAFETY_CALL	CALL F	38	Función
	FC5001	SAFETY_OB	KOP F	128	Función
	FC5002	SAFETY_HARDWARE	KOP F	88	Función
	FC5003	SAFETY_INTERCONEXION	KOP F	398	Función
	FC5005	SAFETY_STOP	KOP F	674	Función
	FC5006	SAFETY_DEVICE	KOP F	126	Función
	FC5011	SAFETY_SUPPLY	KOP F	1482	Función
	DB5004	F_P02_DI	DB F	664	DB de instancia del FB 5000
	DB5005	F_P01_DI	DB F	664	DB de instancia del FB 5000
	DB5006	F04100_4_8_F_DI_DC24V	DB F	664	DB de instancia del FB 5000
	DB5007	F03020_FDI10xDC24V_2A	DB F	664	DB de instancia del FB 5000
	DB5008	F03010_FDI24xDC24V	DB F	664	DB de instancia del FB 5000
	DB5500	Safety_DB	DB F	58	Bloque de datos

Figura 51. Bloques de seguridad del programa

7 PROGRAMA TIA PORTAL

En este último punto y para finalizar, se explicarán brevemente las pantallas desarrolladas en TIA Portal para poder controlar la instalación. Se pretende también relacionar la programación del PLC con la programación de las HMI y a su vez relacionarla con la visualización directa del transporte de carrocerías. Además, se explicará alguna pantalla de configuración de algún parámetro en concreto del proceso y alguna pantalla de avisos. Con esto se pretende hacer tangible todo el trabajo de programación y facilitar la supervisión del transporte por parte de operarios o trabajadores de mantenimiento.

7.1 IMÁGENES DEL PROYECTO

En TIA Portal a cada una de las pantallas de las HMI se les llama imágenes. Seguidamente se explicarán las imágenes que se han considerado más importantes en el desarrollo y cómo se programan.

7.1.1 Imagen 01-Layout Principal

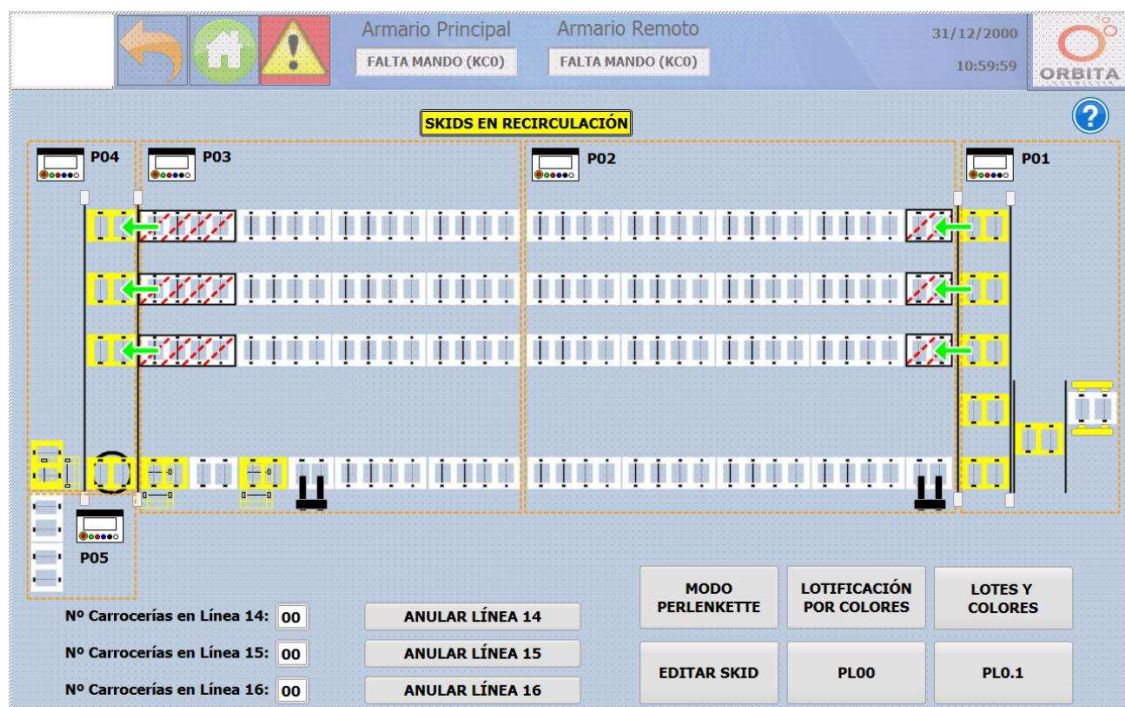


Figura 52. Imagen del Layout principal de la instalación

En la figura 52 se observa la imagen principal de la instalación. Como se puede ver, se encuentran los cinco pupitres y las mesas que controla cada uno de ellos que, en esta vista, no cuentan con todos los componentes, entre los que se encuentran los detectores y los indicadores de motor. Las mesas que realicen algún tipo de movimiento, ya sea de giro o de elevación, se han representado en color amarillo mientras que, el resto de las mesas, se han representado en color blanco. Toda la sección que abarca cada pupitre está cubierta por un botón invisible para activar su imagen correspondiente. En el interior, se encuentran las mesas que controla el pupitre. Estas mesas tienen una variable enlazada, (Mesa_1_XX_X.oMR_Fault) correspondiente a cada mesa para que, en caso de entrar en fallo, parpadee en color rojo. En la figura

53 se puede apreciar que se enlaza el fallo con el cambio de color de la mesa. Cuando el bit está a 0 se visualizará en blanco y cuando este a 1 se verá en rojo. Este bit corresponde al DB del grupo funcional de la mesa que se ha explicado anteriormente.



Figura 53. Fallo en mesa

En la figura 52 también se encuentran unos contadores para controlar el número de carrocerías que se encuentran en cada línea y botones para anular las líneas 14, 15 y 16, de forma que estas no extraigan carrocerías. Cuando estos botones estén activados cambiarán a color verde y aparecerá sobre la última mesa de la línea anulada un gráfico con líneas rojas para que sea muy intuitivo.

También se ha dispuesto el botón de modo perlenkette. Dicho modo consiste en que las carrocerías se introducen a las líneas 14, 15 y 16 de dos en dos de forma cíclica. Además, para la salida de estas líneas a la mesa transfer, también lo harán por parejas. Otra acción que realiza dicho botón es la anulación del modo de trabajo de lotificación por colores si este se encuentra activo.

Al activar el modo de lotificación por colores, primeramente, se desactiva el modo perlenkette. Este modo distribuye las carrocerías por las líneas 14, 15 y 16 en función del color de imprimación seleccionado para cada una de ellas. Los colores posibles son:

- 0190: Color gris.
- 0001: Color blanco.
- 0173: Color antracita.

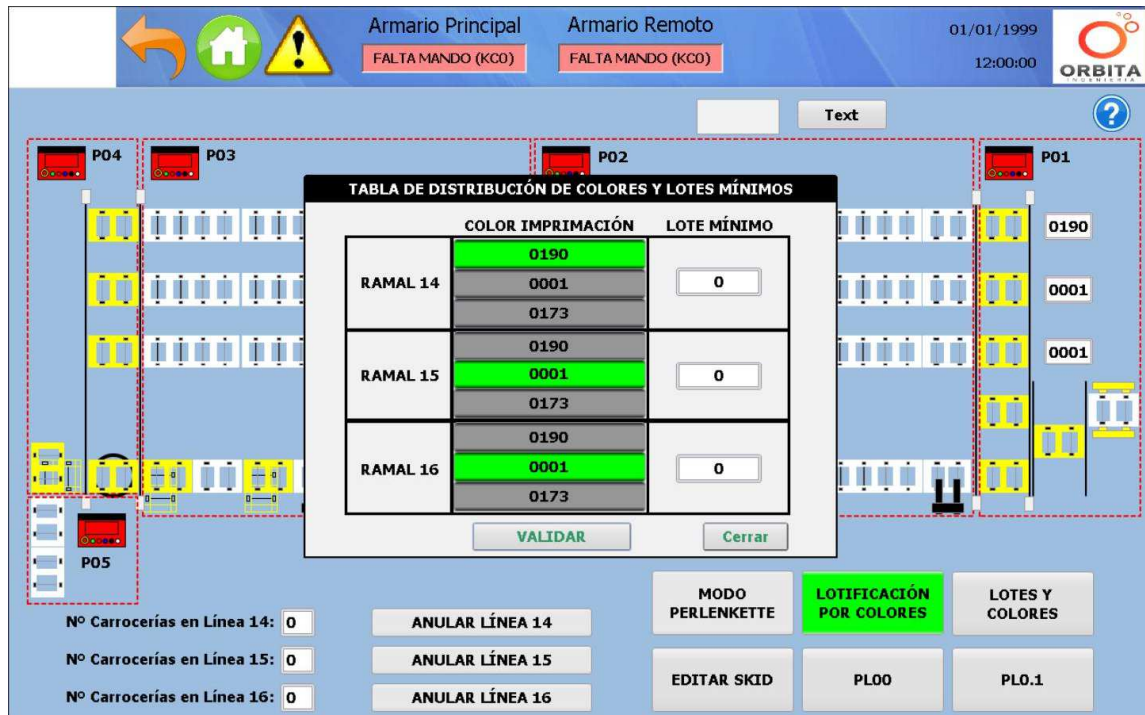


Figura 54. Ventana de lotificador por colores

En la figura 54 aparece la ventana de selección del color de imprimación y el lote mínimo para cada ramal, que se debe encontrar entre 8 y 17. Esta ventana se hace visible cuando se pulsa el botón de lotes y colores, el cual únicamente es visible cuando el modo de lotificación por colores se encuentra activo. Una vez se valida un color de imprimación, se activa la visualización de dicho número al lado del ramal en cuestión.

7.1.2 Imagen 02-Pupitre 02

Como se ha comentado en la imagen principal, pulsando sobre la región que abarca el pupitre 2, se accede a una vista más detallada de las mesas correspondientes a esta zona. En la figura 55 se puede ver que da mucha información del estado de esas mesas. Se aprecia si está ocupada o no con la

figura de un coche, si entra en fallo o no cambiando de color a rojo, si el propio motor está activo o si está en fallo, si los sensores de la mesa detectan, etc.

Cada mesa que se observa se ha creado con un tipo de faceplate en concreto. Un faceplate es parecido a una FB como las que se han empleado en PLC, pero aplicadas en las pantallas. En el siguiente punto se explica con mayor detalle.

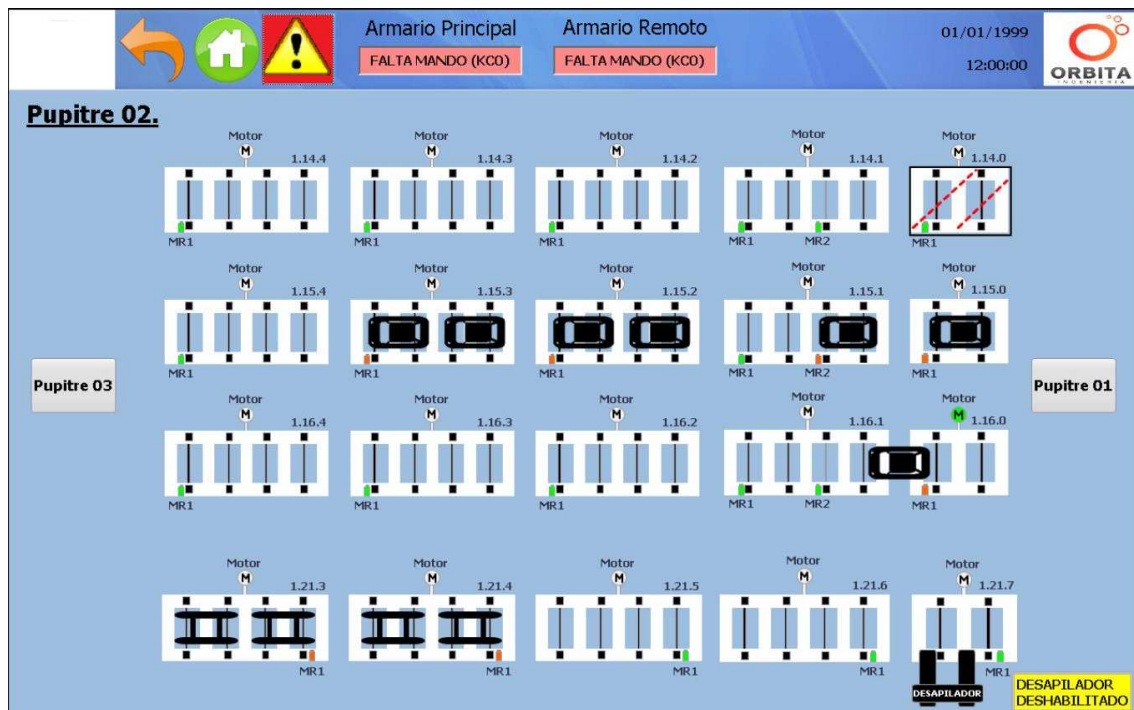


Figura 55. Imagen del Pupitre 2.

7.2 FACEPLATES

Si se pretende definir de forma sencilla lo que es un faceplate, podría definirse como una plantilla de objetos. Es por eso por lo que se les denomina plantilla de imagen. La utilización de faceplates facilita la inserción de nuevos objetos que se comporten del mismo modo, y, por lo tanto, sean susceptibles de realizar con ellos una plantilla. De este modo, una vez se crean, únicamente se tiene que añadir una nueva instancia de esta plantilla y realizar la configuración del intercambio de datos.

Conectándolo con la programación realizada, estos faceplates son muy útiles porque enlazan las variables directamente con los DBs de las mesas, o sea, con el grupo funcional al que corresponde cada mesa. Por eso, generando un tipo de faceplate, se puede utilizar para varias mesas enlazándolo con su DB correspondiente y cambiando el nombre.

7.2.1. Creación de Faceplates

En la figura 56 se puede observar un ejemplo de un faceplate en desarrollo. En TIA Portal se puede trabajar por capas y como se puede apreciar en cada capa se tiene la visualización de un Step Cycle diferente, es decir, un estado de la mesa diferente. Se le da visibilidad a la figura de la carrocería en función del Step Cycle que está activo. A parte, también se animan los sensores y el círculo del motor. Es cómodo para trabajar porque las variables a tratar se han generado con anterioridad a través de los DBs de las mesas y por lo tanto es fácil de enlazar. Estas variables se van actualizando todo lo rápido que va la conexión efectuada por cable ethernet al PLC.

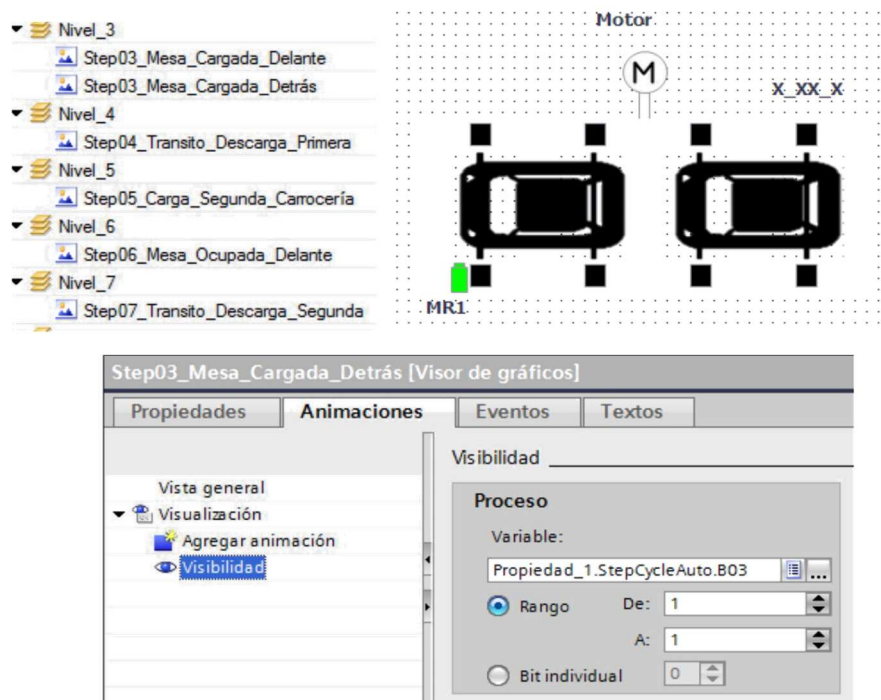


Figura 56. Ejemplo creación faceplate

Aquí concluye la explicación del desarrollo del programa PLC y de las pantallas HMI que conforman la instalación de transporte de carrocerías. Como se puede observar, pese a que los PLC y las HMI han sido programados en entornos distintos (STEP 7 y TIA Portal, respectivamente), su programación está directamente relacionada, y además, en ambos casos se ha trabajado con el fin de lograr una mayor uniformidad en la programación existente en el conjunto de la planta.

8 ANÁLISIS DEL IMPACTO AMBIENTAL

En este apartado se describirá el impacto ambiental que tiene la realización del presente proyecto. Principalmente, los impactos ambientales generados por la instalación considerada son dos:

- Emisión de calor a la atmósfera: esta es una instalación de grandes dimensiones, que cuenta con un número muy elevado de equipos que pierden energía en forma de calor, tales como los motores que accionan el movimiento de las mesas. Mediante el sistema de climatización de la planta se logra mantener una temperatura adecuada en su interior, pero todo este calor extraído del interior es emitido a la atmósfera. Esto puede generar un incremento de la temperatura del entorno, que puede tener efectos perjudiciales para el medio ambiente, tales como el impacto sobre el ecosistema de los alrededores. Sin embargo, cabe destacar que se trata de una zona industrial desde hace décadas, y que la fauna y la flora es escasa, con lo cual el impacto ambiental producido por este motivo es reducido.

Además, estas instalaciones de mesas, motores, etc., eran existentes, y por lo tanto, el impacto ambiental ocasionado se considera fuera del alcance del presente proyecto.

- Generación de residuos eléctricos y electrónicos: Cabe destacar que este impacto no se produce de manera inmediata, si no en el momento en el cual los componentes eléctricos y electrónicos que conforman la instalación dejen de ser útiles y se conviertan en residuos. Es decir, cuando dejen de funcionar o sean sustituidos por otros componentes más modernos. Por lo tanto, se trata de un impacto ambiental a largo plazo.

Los componentes eléctricos y electrónicos pueden contener metales pesados, cadmio, plomo, mercurio... Además, usan diferentes tipos de

plásticos no degradables. Esto puede ocasionar la contaminación en las aguas de subsuelo, ríos, lagos, mares y áreas verdes. Además de contaminar el entorno, puede llegar a provocar daños en el ADN y cáncer, además de inflamación y estrés oxidativo, derivando en enfermedades cardiovasculares.

Para reducir el impacto ambiental causado, es necesaria la reutilización de estos componentes siempre que sea posible, para minimizar la cantidad de residuos generada. En el caso de que la reutilización no fuera viable, se deberá proceder a su correcto reciclaje. El Real Decreto 110/2015 obliga a los productores de aparatos eléctricos y electrónicos a adoptar las medidas necesarias para que los residuos de estos aparatos, puestos por ellos en el mercado, dispongan de sistemas de recogida selectiva y tengan una correcta gestión medioambiental.

El proceso para el reciclaje de los componentes eléctricos y electrónicos se describe brevemente a continuación:

1. Recepción y almacenamiento.
2. Clasificación de los equipos.
3. Desmontaje manual y separación de componentes peligrosos.
4. Trituración de materiales valorizables.
5. Separación de materiales y expedición para su valorización externa.

Debido a que este tipo de productos están formados por una gran cantidad de componentes de distintos materiales, su reciclaje implica un esfuerzo importante. Además, la valorización de determinados componentes irremediablemente generará contaminación, como, por ejemplo, la emisión de toxinas a la atmósfera.

A modo de resumen del análisis del impacto ambiental, se muestra la tabla 1.

Tabla 1. Resumen resultados análisis impacto ambiental

Impacto ambiental	Impacto proyecto	Efecto	Gravedad impacto	Acciones a realizar	Gravedad tras las acciones
Emisión de calor	Indirecto	Inmediato	Baja	-	Baja
Residuos electrónicos	Directo	A largo plazo	Alta	Reciclaje	Moderada

Se concluye que este proyecto implica un impacto ambiental directo, que es la generación de residuos eléctricos y electrónicos. Pese a ser un problema a largo plazo, esto supone un impacto ambiental grave, que puede ser mitigado mediante el reciclaje de estos equipos, aunque sus efectos no podrán ser eliminados por completo.

9 VALORACIÓN ECONÓMICA DEL PROYECTO

En este apartado se presenta el presupuesto del proyecto realizado. Es importante señalar que este presupuesto únicamente contempla las horas de ingeniería electrónica y asistencia. Por tanto, no se consideran los costes del diseño de la instalación de equipos y controles, dado que este proyecto se ha enfocado únicamente en el desarrollo de una solución de ingeniería y en su programación.

A continuación se muestra la tabla 2, que contiene el presupuesto con el desglose de horas. La valoración económica se ha realizado teniendo en cuenta un precio medio por hora de 25€.

Tabla 2. Presupuesto de la parte de programación del proyecto

TAREA		HORAS	COSTE
Análisis funcional		50	1.250€
Programa PLC	Desarrollo código	200	6.875€
	Simulación en oficina	75	
Programa HMI	Desarrollo pantallas	90	3.200€
	Simulación en oficina	38	
Puesta en marcha	Simulación en planta (estado OFF)	526	19.650€
	Verificación (ON controlada)	260	
Asistencia a la producción		490	12.250€
Capacitación cliente		8	200€
Manuales y documentación		40	1000€
Puntos pendientes		36	900€
Actuaciones en garantía		12	300€
TOTAL		1175	45.625€

El proyecto se inicia con un análisis funcional, que tiene el objetivo principal de definir las funciones que serán necesarias para cumplir los requerimientos funcionales, es decir, para que la instalación de transporte de carrocerías

realice todas las acciones que se desean: inicio de ciclo, carga, descarga, paradas de emergencia, operación en modo manual, operación en automático, comprobación de fallos, etc. Este es un proceso previo al desarrollo del código, y pese a que el número de horas no es muy elevado, ya que solo se definen las funciones a nivel conceptual, es muy útil y necesario para sentar las bases del programa.

La fase de desarrollo del programa se lleva a cabo en oficina. A partir del funcional, se escribe el código, empleando el lenguaje KOP, necesario para cumplir todas las funciones que se requieren para que la instalación opere adecuadamente. Esta es la programación de los PLC, y es una de las fases con mayor inversión de horas.

Una vez desarrollado el programa de los PLC, se realiza la programación de las pantallas HMI. Estas, pese a utilizar un entorno de programación distinto al de los PLC, siguen la misma lógica, y por lo tanto el número de horas necesarias se reduce respecto a las empleadas en la fase anterior. Tanto el desarrollo del programa de PLC como el de las pantallas, cuentan con unas horas dedicadas a la simulación, que se puede realizar en la propia oficina mediante el programarlo ofrecido por Siemens.

Con el programa desarrollado, es momento de su implantación o puesta en marcha. Es una de las fases con mayor inversión de horas, y se divide en dos etapas diferenciadas. Hay una fase previa de simulación en planta en estado OFF, es decir, cuando la instalación no se encuentra produciendo. Esto es muy útil para encontrar posibles errores que no hayan sido detectados durante el desarrollo del código o durante la simulación en oficina.

Posterior a esas simulaciones en OFF, se empieza a producir. Esta es una etapa de verificación, en la cual ya ha arrancado la producción, pero de manera controlada, para poder solventar cualquier incidencia que se produzca.

Superada la fase de puesta en marcha, se dedica una gran cantidad de horas a la asistencia a la producción. En esta etapa, la instalación ya está en completo funcionamiento, y por tanto, debe ser capaz de alcanzar el ritmo de producción deseado. El cliente solicita una gran cantidad de horas de asistencia para asegurar que todo funcionará correctamente, y que cualquier fallo podrá ser solventado de manera rápida, para alterar lo mínimo posible la eficiencia productiva. Además, en esta fase también se realizan modificaciones que el propio cliente solicita, después de haber visto la instalación en pleno funcionamiento, con el objetivo de mejorar todo cuanto sea posible antes de entregar la instalación definitivamente.

Es necesario dedicar unas horas a la capacitación del cliente, es decir, a la formación de operarios e ingenieros de la planta, para darles las instrucciones básicas de cómo funciona la instalación. Esta formación tendrá especial importancia sobre todo a nivel del funcionamiento de las pantallas HMI, ya que la pantalla es el medio de comunicación entre el operario y las mesas de transporte de carrocerías.

Estas instrucciones de funcionamiento se documentan y se entregan al cliente en forma de manuales. Estos tienen el objetivo de describir el funcionamiento completo de la instalación, definiendo todos los elementos, señales, modos de funcionamiento, etc., de manera que el cliente disponga de toda la información para poder solventar cualquier fallo o avería que se produzca en el futuro.

Se emplean unas pocas horas en la resolución de puntos pendientes, es decir, cuestiones que no se han podido completar en otras fases, o que han surgido posteriormente.

Por último, se dedican unas horas a las actuaciones en garantía, que hacen referencia a la corrección de errores que han podido surgir en las últimas fases de la puesta en marcha y que requieren de pequeñas intervenciones que permitan acabar de pulir el programa y aumentar la robustez de la instalación.

Teniendo en cuenta el total de horas empleadas, y considerando que esas horas se reparten entre 2 programadores, en jornadas de 8 horas al día, la duración es de aproximadamente 70 días laborables, que equivale a unos tres meses y medio dedicados al proyecto. El coste asociado a esos trabajos es de 45k€. Por lo tanto, se observa que se trata de un proyecto de gran envergadura, que requiere de muchas horas de dedicación y de un coste considerable, solo en lo que respecta a la programación. Evidentemente, el coste del global del proyecto, contando las fases de ingeniería eléctrica, el diseño y el montaje de equipos, etc., supone una inversión mucho mayor que la especificada en este presupuesto.

10 CONCLUSIONES

Una vez finalizado este proyecto, se puede concluir que se han cumplido satisfactoriamente los objetivos que se fijaron en su inicio.

A partir de la realización de este proyecto, ha sido posible reducir el tiempo de proceso de las estaciones de transporte en varias líneas de fabricación de automóviles. Esto permite producir un mayor volumen de producto en menos tiempo, lo cual se traduce en un aumento de la eficiencia del proceso productivo.

Este incremento en la productividad se ha logrado mediante el desarrollo de un nuevo programa de PLC: se han generado funciones más eficientes y se ha creado un programa acorde a las capacidades de los nuevos PLC instalados y a las necesidades productivas.

Otro de los objetivos más importantes de este proyecto era la estandarización de la programación de la planta. Toda la programación se ha llevado a cabo en STEP 7 de SIEMENS, y todas las pantallas de los paneles de operador HMI se han programado con el entorno TIA Portal, también de SIEMENS.

También con el objetivo de unificar toda la programación, se han generado las funciones FB. Las FB han sido utilizadas para crear una única función que se puede ejecutar en distintas ocasiones, con el objetivo de agilizar la programación. Por lo tanto, se han desarrollado funciones estándar, cada una de ellas adecuada a las necesidades de cada tipo de mesa de transporte.

El uso de las FB ha permitido establecer un modelo para el desarrollo de código en futuras intervenciones en la planta. Con las FB se podrá programar cualquier instalación de características similares, adaptando este modelo a los requerimientos de cada una de las tipologías de las mesas. Esto se podrá llevar a cabo a partir de mínimos cambios en el programa modelo de partida. El uso de este modelo servirá para reducir el tiempo de desarrollo de nuevos

programas y facilitar la comprensión del código a cualquier técnico que tenga que desarrollar o acceder al programa.

Debido a que las condiciones y limitaciones estaban marcadas por los elementos mecánicos existentes en la propia instalación (sensores, actuadores, motores, etc.) se han estudiado todos los elementos y se ha desarrollado la actualización de los autómatas mediante la modificación únicamente a nivel eléctrico y de programación. Por lo tanto, ha sido posible combinar la tecnología preexistente en la planta, con las nuevas técnicas implantadas, para lograr una mejora de las condiciones de funcionamiento del sistema de transporte de los automóviles.

Se ha realizado un análisis del impacto ambiental asociado a este proyecto, y se ha visto que la principal problemática es la generación de residuos electrónicos. Sin embargo, este impacto se producirá a largo plazo, cuando los equipos sean sustituidos, y podrán minimizarse sus efectos mediante el correcto reciclaje de esos residuos.

En cuanto a la valoración económica, este proyecto supone una inversión de aproximadamente 45.000€, únicamente teniendo en cuenta los trabajos de programación y asistencia efectuados. Pese a que la inversión global del proyecto será mucho mayor, este valor permite constatar que se trata de un proyecto de gran envergadura, que ha requerido de muchas horas de trabajo, tanto en oficina como en planta, para su total realización.

11 BIBLIOGRAFÍA Y REFERENCIAS

[1] Controlador lógico programable. En: Wikipedia [en línea]. Fundación Wikimedia, 2019. [Consulta: 10 febrero 2019]. Disponible en:
<https://es.wikipedia.org/wiki/Controlador_l%C3%B3gico_programable>

[2] Aplicaciones de los PLC en la Industria Moderna. En: Aldakin [en línea]. Aldakin, 2019. [Consulta 15 febrero 2019]. Disponible en:
<<http://www.aldakin.com/aplicaciones-plc-industria-moderna/>>

[3] Estructura Interna del PLC. En: Google Sites [en línea]. Google Sites, 2018. [Consulta 3 marzo 2019]. Disponible en:
<<https://sites.google.com/site/20131036eq4/analizas-las-caracteristicas-y-estructura-de-los-plc/estructura-interna-del-plc>>

[4] Ciclo de trabajo de un PLC. En: SlideShare [en línea]. González, S., 2009. [Consulta 3 marzo 2019]. Disponible en:
<<https://es.slideshare.net/hustlergs/ciclo-de-trabajo-de-un-plc>>

[5] STEP 7. En: Wikipedia [en línea]. Fundación Wikimedia, 2019. [Consulta 3 marzo 2019]. Disponible en:
<https://es.wikipedia.org/wiki/STEP_7>

[6] Lenguaje de programación KOP. En: www.grupo-maser.co [en línea]. Grupo Maser, 2019. [Consulta 10 marzo 2019]. Disponible en:
<http://www.grupo-maser.com/PAG_Cursos/Step/step7/Proyecto%20step7/paginas/contenido/step7/7/2.9.1.2.htm>

[7] SCL Lenguaje estructurado de Siemens. En: www.tecnopl.com [en línea].
Tecnopl, 2019. [Consulta 10 marzo 2019]. Disponible en:

<<http://www.tecnopl.com/scl-lenguaje-estructurado/>>

[8] STEP 7, AWL, FUP, KOP. En: www.programacionsiemens.com [en línea].
Gútiérrez, I., 2013. [Consulta 10 marzo 2019]. Disponible en:

<<https://programacionsiemens.com/step-7-awl-fup-kop-cual-elijo/>>

[9] SIMATIC STEP 7 Professional (TIA Portal). En: www.w5siemens.com [en
línea]. Siemens.com Global Website, 2019. [Consulta 15 marzo 2019].
Disponible en:

<<https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/tia-portal/controller-sw-tia-portal/simatic-step7-professional-tia-portal/pages/default.aspx>>

[10] Bloques organizacionales OB - PLCs. En: www.autracen.com [en línea].
Autracen, 2019. [Consulta 15 marzo 2019]. Disponible en:

<<http://www.autracen.com/bloques-organizacionales-ob-plcs/>>

[11] FC en STEP 7. En: www.programacionsiemens.com [en línea].
Gútiérrez, I., 2013. [Consulta 15 marzo 2019]. Disponible en:

<<https://programacionsiemens.com/db-en-step-7-operaciones-iii/>>

[12] SIMATIC STEP 7 Safety (TIA Portal). En: www.w5siemens.com [en línea].
Siemens.com Global Website, 2019. [Consulta 17 marzo 2019]. Disponible en:

<<https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/tia-portal/controller-sw-tia-portal/simatic-step7-safety-advanced-tia-portal/pages/default.aspx>>

[13] DB en STEP 7 - Operaciones. En: www.programacionsiemens.com [en línea]. Gútiez, I., 2013. [Consulta 17 marzo 2019]. Disponible en:

<<https://programacionsiemens.com/db-en-step-7-operaciones-iii/>>

[14] Cómo observar y forzar variables en STEP 7. En: www.programacionsiemens.com [en línea]. Gútiez, I., 2018. [Consulta 17 marzo 2019]. Disponible en:

<<https://programacionsiemens.com/como-observar-y-forzar-variables-en-step-7/>>

[15] Software TIA Portal. En: www.w5siemens.com [en línea]. Siemens.com Global Website, 2019. [Consulta 18 marzo 2019]. Disponible en:

<<https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/pages/tiaportal.aspx>>